SAPIENZA
UNIVERSITÀ DI ROMA

Supporting Situated Spoken Human-Robot Interaction through Perceivable Context

Andrea Vanzo

ID number 1642571

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Engineering in Computer Science

XXX Cycle

Department of Computer, Control, and Management Engineering
Sapienza University of Rome
Rome, Italy

April 2018

Thesis Advisor

Prof. Daniele Nardi

Co-Advisors

Prof. Roberto Basili
Prof. Tiziana Catarci

Review Committee

Dr. Mary Ellen Foster
Prof. Manfred Tscheligi

Thesis defended on September 7, 2018
in front of a Board of Examiners composed by:

Prof. Paolo Boldi (chairman)

Prof. Fabio Massimo Zanzotto

Prof. Sven Wachsmuth

Supporting Situated Spoken Human-Robot Interaction through Perceivable Context

Keywords: Spoken Human-Robot Interaction, Context-aware interaction, Natural Language Processing, Natural Language Understanding

Ph.D. thesis. Sapienza University of Rome

Version: September 3, 2018

Website: https://andreavanzo.gitlab.io/

Author's email: vanzo@diag.uniroma1.it

*Here is no choice but either **get results** or die.*
Sir William Wallace
(. . . more or less. . .)

Abstract

At a growing pace, the presence of robots in everyday environments is increasing day by day. In fact, their incredibly wide applicability, spanning over various environments and scenarios, is speeding up such spreading. Industrial and working environments, healthcare assistance in public or domestic areas are highly benefiting from robots' services, that are able to accomplish manifold tasks that have become difficult and annoying for humans. As an example, in domestic environments robots can be deployed in a vast plethora of applications, supporting humans in everyday activities. However, robots are not yet comparable to humans in terms of reasoning and autonomy: a complete knowledge of the environment robots are deployed into is often required to both accomplish the desired task and effectively improve the interaction experience with the user. In this perspective, an active interaction with the end user is still a valuable solution for alleviating this lack of autonomy, as in the so-called Symbiotic Autonomy.

My thesis analyzes the impact of such a contextual knowledge in several Human-Robot Interaction (HRI) sub-tasks, with a particular attention on when information, desiderata, and knowledge are shared through natural language. In fact, natural language can be considered one of the most natural way of communicating. To this end, three HRI-specific problems have been considered. First, the importance of the environmental context has been analyzed in a scenario where the robot is not able to achieve its tasks on its own and it needs to ask humans for help. The thesis will address how some perceivable characteristics of the environment might help in designing the robot's behaviors. The second scenario refers to the ability of re-ranking the transcriptions hypothesized by an Automatic Speech Recognition (ASR) in a situated command interpretation task. This dissertation will show that context, encoded as domain-dependent information, can actually improve the accuracy of a free-form domain-independent ASR. The third scenario relates to the task of interpreting robotic commands, where the robot has to react to commands expressed through natural language. This thesis will provide evidence that a structured representation of the environmental knowledge is beneficial for coherently mapping the sentence to the correct interpretation in a situated scenario. The last scenario investigates to what extent, when acquiring the above mentioned structured representation of the environment through a dialogic guidance, an active perception of the environment improves the dialogic experience, by decreasing the tutoring cost. In this respect, different types of context have been considered and defined for each scenario, ranging from information that is actively perceivable and observable by the robot, to structured knowledge acquired through pre-processing stages.

Acknowledgments

Sometimes it just takes a little longer to get to your destination, but if you make sure to enjoy the journey...

During my journey, I had the honor to work with brilliant researchers, and to be supported by wonderful people.

First, I want to thank my advisor, Daniele Nardi. His passion for research, teaching, and dedication to young students has been of inspiration for my growth as a researcher. I enjoyed each and every single moment spent together, chatting on interesting research problems, traveling all around the world to show our works and going biking through mountain paths.

Then, I would like to thank my co-advisor Roberto Basili and Danilo Croce. Actually, my journey started thanks to their passion, encouragement, and support. I really enjoyed the 8 years working together, full of discussions (till late night) to improve our research and understand this wonderful world that is Natural Language. Without your dedication, I would not have been who I am.

I am grateful to Oliver Lemon who gave me the opportunity to enrich my journey with a wonderful experience and to keep working in his lab.

Thanks to all the Lab.Ro.Co.Co. members, for having been more than simple colleagues during these three years. In particular, I want to thank Francesco: we started our journey together, sharing ideas, problems, and pink rooms. I am grateful to the SAG members, in particular, Giuseppe and Emanuele, for having shared their passion, knowledge, and friendship with me. Thanks to all the Interaction Lab members. Someone else said that *friendship is born at the moment when one man says to another: What! You too? I thought that no one but myself...* Thanks, Christian and Yiannis for your friendship and the 6 months spent together.

Thanks to all my friends in Cittareale and Rome, in particular, Giorgio, Andrea, Valerio, Tiziano, Andrea, and Paolo: you all have been the anchor that kept my feet in the real world, contributing to shape the person I am now.

Thanks to Flavia, for being such a wonderful partner since more than 10 years. In you, I found support, comprehension, friendship and, definitely, the love of my life. We grew up together and I hope we will become old together.

Last but not least, thanks to my family, my father Angelo, my mother Fiorella and my sister Valentina. I could not have wished better for my life: you have been silently supporting me throughout my life, always indicating to me the right path to follow. If I reached the destination it is only thanks to you.

This thesis is dedicated to Matteo and Martina: may all your dreams come true.

...eventually you get there.

Contents

List of Figures

List of Tables

Acronyms

AC Argument Classification. 65–67, 69

AI Artificial Intelligence. 22, 129

AIML Artificial Intelligence Markup Language. 30, 78–80

AMR Abstract Meaning Representation. 81

ASR Automatic Speech Recognition. v, 7, 10, 13, 17, 30, 39–44, 46–49, 77, 81, 99, 105

BI Boundary Identification. 63–68

CCG Combinatory Categorial Grammar. 17

CFG Context-Free Grammar. 17

CFR Command Frame Representation. 79, 81

CNN Convolutional Neural Network. 85, 89, 155

CRF Conditional Random Field. 17, 53

DM Dialogue Manager. 9, 19, 30, 78, 79, 85, 87, 89, 101, 106

DS Distributional Semantics. 18, 21, 51, 53, 54, 59, 65–67, 69, 70

FD Frame Detection. 61–67

FSG Finite State Grammar. 41

GUI Graphical User Interface. 30

HAM Human Augmented Mapping. 9, 15, 17, 22, 23, 56, 75, 85, 86

HMM Hidden Markov Model. 140, 141, 143, 145

HRI Human-Robot Interaction. v, 1, 2, 4, 6, 7, 11, 13, 15–19, 21, 22, 28, 38, 39, 42, 48, 55, 56, 81, 83, 103, 104, 155

HuRIC Human-Robot Interaction Corpus. 8, 40, 46, 66, 71, 73, 75, 80, 84, 106

IFR International Federation of Robotics. 1

KB Knowledge Base. 30, 78–80

KeLP Kernel-based Learning Platform. 61, 66, 81

LB-SOINN Load-Balancing Self-Organizing Incremental Neural Network. 9, 85, 89, 99, 100, 106, 155, 156

LU4R adaptive spoken Language Understanding chain For Robots. 9, 52, 75–81, 83, 84

MDP Markov Decision Process. 9, 12, 85, 91, 92, 97, 100, 106, 107, 146–149

ML Machine Learning. 8, 11–13, 17, 52, 60, 100, 105, 106, 129, 130, 150

NL Natural Language. 2, 6, 10, 13, 15–19, 42, 48, 51, 71, 81, 84, 89, 103–105, 107

NLG Natural Language Generation. 89

NLP Natural Language Processing. 129

NLU Natural Language Understanding. 41, 49, 52, 89

NP Noun Phrase. 55

PAC Probably Approximately Correct. xiv, 130–132

POMDP Partially Observable Markov Decision Process. 147

POS Part-Of-Speech. 62, 64, 65, 71, 74, 80, 143

PP Prepositional Phrase. 8, 51, 53, 55, 68

REG Referring Expression Generation. 18

RL Reinforcement Learning. xv, 9, 12, 13, 85, 88, 89, 91, 92, 97, 100, 106, 146, 148, 149

ROS Robot Operating System. 77–80

SARSA State-Action–Reward-State-Action. 97, 149

SHRI Spoken Human-Robot Interaction. 13, 15, 17, 18, 49, 84, 101, 103–107, 130

SKB Support Knowledge Base. 77, 79

SLM Statistical Language Model. 41

SLU Spoken Language Understanding. 56, 75–77, 80

SOINN Self-Organizing Incremental Neural Network. 155

SRL Semantic Role Labeling. 53

SVM Support Vector Machine. xiv, 17, 60, 61, 81, 133, 137, 138, 143–145, 155

SVM^{hmm} Hidden Markov Support Vector Machine. 8, 51, 60, 61, 66, 81, 106, 145, 146

SVM^{struct} Structural Support Vector Machine. 143, 145

VP Verb Phrase. 55

XDG eXtended Dependency Graph. 81

Preamble

Robotics has always fascinated humans. Robots have been part of the popular culture since the dawn of time, even before a clear understanding of what a robot is, stimulating humans' fantasy and dreams. Such attraction has been motivated by the mental representation of the robot, rather than the robot itself, intended as a machine. Humans are continuously representing the robot as an expression of its being and capabilities. The idea of creating *something* that is able to reproduce human's behaviors and dynamics pushed the research to get where we are today. In fact, robots are really becoming *"alive"*, in the sense that, what we have been imagining for years (or a good approximation of it) is eventually getting out of academia and research centers and becoming part of our everyday life.

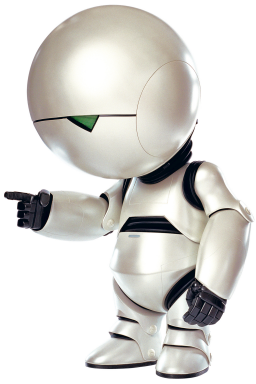
On the one hand, it is true that robotics is the timeless humans' dream; on the other hand, it is interesting to contextualize robotics within the popular culture, to understand what the end user is expecting from such systems. Among others, sci-fi cinema and literature are the arts that mostly contributed to creating such a dream of robots and that established questions and challenges researchers are still trying to find a solution to.

One of the most famous examples of an Artificial Intelligence is *HAL 9000*, from Kubrick's *"2001: A Space Odyssey"*. There is a quote that is particularly significant to us, as it poses several questions about what a robot is expected to do:

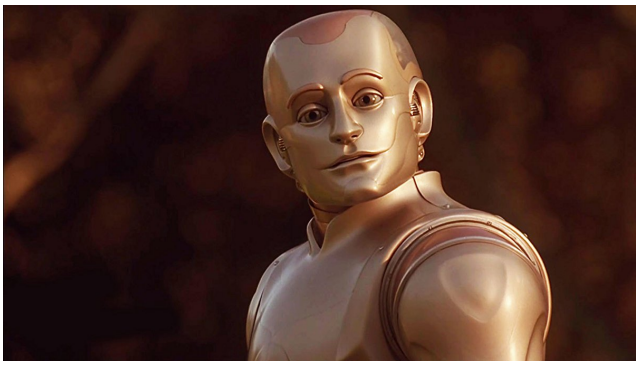
> **HAL 9000**: "I'm afraid. I'm afraid, Dave. Dave, my mind is going. I can feel it. I can feel it. My mind is going. There is no question about it."
>
> (*2001: A Space Odyssey – 1968*)

Such excerpt provides several hints for reflection. First, *"I'm afraid"* and *"I can feel it"* suggest a specific feature for an artificial being: the ability to feel something, to perceive sentiment and react accordingly. Is it possible for a robot to manifest its feelings and thoughts? Do we really need robots that are able to enter our inner life and operate on it? Second, when HAL 9000 says *"My mind is going"* it means it is aware of itself, capable of identifying and understand its limitations and state.

If you are reading this thesis, you probably know *"The Hitchhiker's Guide to the Galaxy"*, a sci-fi book written by the visionary writer *Douglas Adams*.



(a) Marvin, the Paranoid Android - The Hitchhiker's Guide to the Galaxy (2005)



(b) Andrew - Bicentennial Man (1999)

Figure 0.1. Example of robots in cinematography

> **Marvin**: "Here I am, brain the size of a planet, and they ask me to take you to the bridge. Call that job satisfaction, 'cause I don't."
>
> (*The Hitchhiker's Guide to the Galaxy – 1979*)

Here, the depressed service robot *Marvin* (Figure 0.1(a)) complains about the way humans make use of it. Thanks to its infinite computational capabilities, it would be able to do more complex tasks than simply escorting people to places, and this is the reason for its depression. In this case, the robot is supposed to be an extremely powerful cognitive system, that is able to accomplish complex computational tasks.

Another example is the robotic main character of the *"Bicentennial Man"* (Figure 0.1(b)). The robot "Andrew" (Robin Williams) is introduced into a family home to perform housekeeping and maintenance duties and, at some point, the patriarch of the family "Sir" teaches Andrew how to tell jokes:

> **Sir**: "Why did the chicken cross the road?"
> **Andrew**: "One does not know, sir, possibly a predator was behind the chicken, or possibly there was a female chicken on the other side of the road if it's a male chicken. Possibly a food source, or depending on the season it might be migrating. One hopes there's no traffic."
> **Sir**: "To get to the other side."
> **Andrew**: "To get to the other side. Ah, why is that funny?"

Figure 0.2. C-3PO - Star Wars (1983)

(*Bicentennial Man – 1999*)

In this case, the robot is a home assistant, supposed to help humans in domestic tasks and to entertain people.

Cinematography and, in general, popular culture are having high and diverse expectations of what robots are supposed to do. Sometimes the robot is seen as a sentimental entity, that is able to perceive and replicate emotions, sometimes as an extremely robust and powerful system, capable to perform very complex tasks that would be impossible for humans to accomplish. However, there is a leading thread that connects all the above fictional interactions (and the usual representation of an AI): the way robots communicate with humans. In the above examples (but, in general, every time humans refer to cognitive systems), humans do not play with some synthetic user interfaces to interact with the system. Robots are intelligent systems that, in turn, do not reply with some special Morse code or through an artificial communication protocol. Hence, robots *must* be provided with the capability of understanding and speaking the Natural Language. For a robot, the interaction capability is so much essential that the lively imagination of George Lucas created an extreme example of a robotic platform, *"C-3PO"* (Figure 0.2), whose main feature is the ability to speak *"over six million forms"* of communication:

Luke: "Do you understand anything they're saying?"
C-3PO: "Oh, yes, Master Luke! Remember that I am fluent in over six million forms of com—"
Han Solo: "What are you telling them?"
C-3PO: "Hello, I think. I could be mistaken."

(*Star Wars Episode VI: Return of the Jedi – 1983*)

Thanks to its ability of understanding and speaking so many languages, C-3PO is often used as an interpreter to other forms of life.

Robotics has always fascinated humans, as well as the idea of enabling robots and intelligence systems of understanding and speaking (at least) the natural language has always fascinated me. For this reason, I decided to commit my research in contributing to this goal.

Chapter 1

Introduction

Robots are rapidly becoming part of our everyday environments, ranging from industrial to domestic ones, where they are expected to support human activities in everyday scenarios, by interacting with different kinds of user. In the last decade, several robotic platforms have been marketed, ranging from vacuum cleaners to industrial or domestic robots. Due to the wide applicability, even the most important IT companies are investing in developing platforms and tools to support the spread of robots in our homes. The International Federation of Robotics (IFR) has just released their two annual World Robotics 2017 reports covering 2016 results. The IFR estimated that sales of all types of service robots for domestic tasks, e.g., vacuum cleaning, lawn mowing, window cleaning, could reach almost 32 million units in the period 2018-2020. Among them, the *iRobot Roomba* (Figure 1.1) is probably one of the best examples, as for its spread in the market and the innovation it brought. It is a vacuum cleaner, able to autonomously navigate the environment, map it and properly plan the cleaning of our homes. Other examples are the Softbank Robotics products (Figure 1.2) currently available both as off-the-shelf tools and for research purposes in manifold activities and tasks (e.g., welcoming customers in a mall [57, 81], supporting teachers during classes [102] or playing soccer [85]).

Human-Robot Interaction (HRI) is a field of research that aims at designing robotic systems that are meant to support humans in everyday tasks. Interaction, by definition, requires communication, that can take several forms. However, in all the above contexts, an essential feature the robot must exhibit is the ability of

Figure 1.1. The iRobot Roomba

Figure 1.2. Softbank Robotics family: Nao, Romeo and Pepper (left to right). Photo © SoftBank

understanding and speaking the humans' language. Cognitive systems are expected to efficiently interact with others through the humans' preferred mode of communication: **Natural Language (NL)**. Arguably, NL is one of the most effective way of communicating: it is hands-free, users do not need a special training and it is expressive and efficient. Designing robots with a proper and effective NL interface would enable such devices to be used by untrained users as, for example, elderly people or children.

However, humans and robots do not speak the same language. In fact, as displayed in Figure 1.3, while speaking and understanding NLs is an innate cognitive ability for humans, this is not the true for robots, where a complete stack of computational processes are required to map NL communications to binary code and enable effective interactions. The robot must be able to respect humans' social rules while interacting. Furthermore, whenever the user utters a sentence, the audio signal must be properly mapped into computational structures manageable by the robot; hence, whenever such signals have been recognized, they have to be *understood*, in the sense that the user real intents must be extracted from the sentence. Finally, according to all the information collected, the robot must be able to plan a coherent response, mapping its internal structures to meaningful NL sentences.

This thesis tackles a specific problem that arises when designing a cognitive robot with NL communication capabilities: the role of the **contextual information** in HRI. In fact, NL communication acquires a specific nature when applied to HRI. Linguistic interactions are context-aware as both the user and the robot access and make references to the environment, i.e., perceived entities of the real world [36]. Moreover, people are biased by the surrounding environment and such interference is reflected into the communicative processes.

This thesis argues that:

1. the interplay between context and interaction in NL is motivated by different reasons, generating different forms of contextual knowledge;

2. different forms of contextual knowledge can be exploited to improve the individual HRI sub-tasks (Figure 1.3).

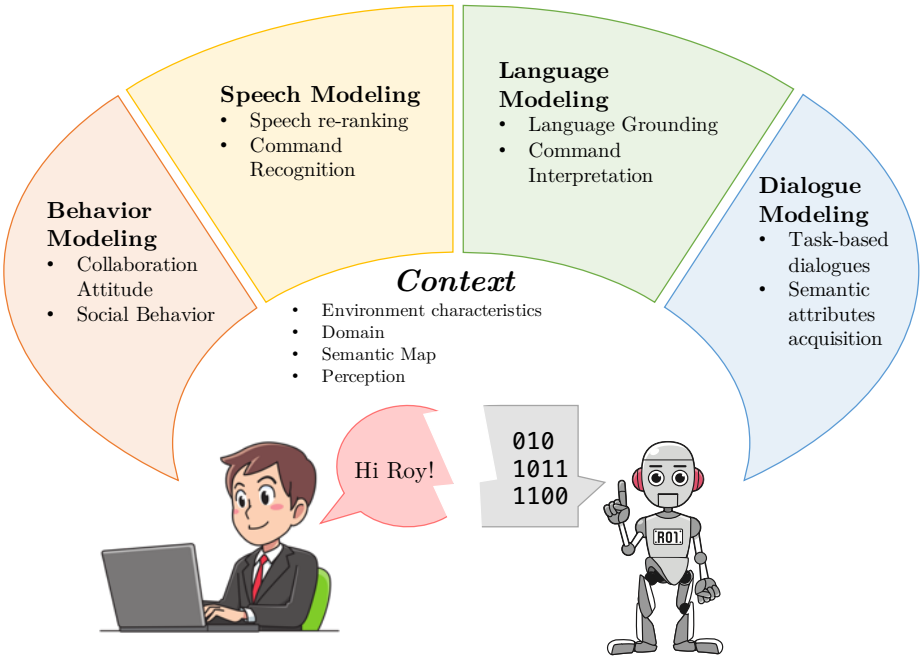


Figure 1.3. Humans and robots do not speak the same language. A full collection of processes is required to enable an effective communication between the two actors. Moreover, interactions are context-aware and the above processes must take into account the role of the operational context, in order mimic humans cognitive processes.

First, context should be carefully taken into account when designing robots behavior to maximize users' expectations and task achievement. In fact, the operational context introduces biases on the way people interact with robots. Robots, in turn, should be able to detect the context and properly exploit such information to optimize their social behavior. For example, a person that is performing a task might not be willing to interact with a robot.

Second, the correct transcription of user's vocal input highly depends on the operational domain. In fact, whenever such information is available, the robot should be able to filter out implausible speech hypotheses through domain-dependent evidence, thus allowing more appropriate transcriptions to emerge from the list of hypotheses.

Third, language significantly interplays with context in the sense that meaning must adhere to the physical world: the interpretation of an utterance is strongly interlaced with the perceived context, as pointed out by psycho-linguistic theories [160]. A correct interpretation is thus more than a linguistic mapping from an audio signal (e.g., the user's utterance) to a meaning representation formalism. Correctness implies physical coherence and the contextual environment must be observable and observed.

Finally, the interaction flow is strictly interlaced with the context as well. Enabling the robot to reason and make inferences about the environment is essential to improve the interaction experience. For example, context might help in minimizing the tutoring cost (i.e., number of dialogue turns required to fulfill the goal of a task-based interaction), providing guesses for a teaching task.

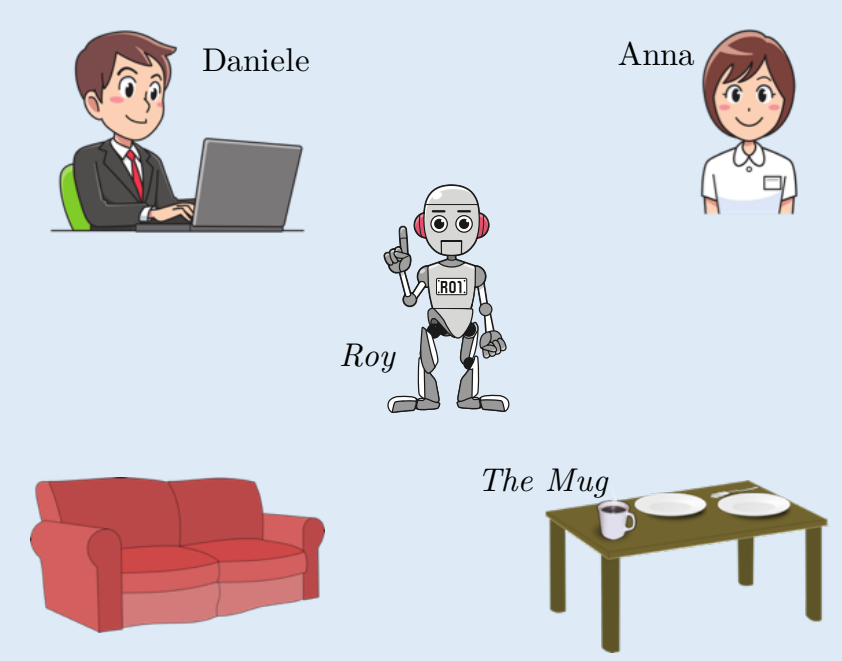


Figure 1.4. The operating scenario: Daniele, Anna, Roy and "The Mug" living in the new home

This introduction is structured as follows. Section 1.1 depicts a working scenario, that helps in contextualizing the contributions developed by this thesis. In Section 1.2 the concept of Situated Interaction is presented, along with a detailed discussion of three HRI sub-problems that have been addressed in our work. Section 1.3 analyzes the contributions of the thesis, providing a list of publications that contributed in drafting this thesis. Finally, Section 1.4 provides a reader-friendly structure of the thesis organization.

1.1. Working Scenario

As a motivation of the work, this section presents an operating scenario, in which the contribution of the context is emphasized with respect to the addressed tasks. Daniele and Anna just moved to a new place (Figure 1.4). They decided to buy a service robot, called *Roy*. Roy is a robotic platform that is meant to execute simple tasks in a domestic environment, such as manipulating objects with its arms, perceiving the surrounding environment through its RGB-D camera, as well as recognizing objects and entertaining humans with simple, but effective interactions. To properly execute the requested commands and reason about the environment, Roy needs a structured representation of its world (see Section 2.4) that is, in this case, the new home of Daniele and Anna [126].

As Daniele and Anna just moved to the place, Roy needs to acquire the representation of the new home. So, during the first tour of the new house, Roy follows Daniele and Anna and it starts building its metric map, attaching semantic information of the environment. In doing this, Roy is allowed to leverage the help of the

users (see Section 2.5.1), by asking them for information of the objects populating the environment. The process of building this map is indeed a combination of its perception capabilities along with humans' feedbacks acquired through natural language interactions. These feedbacks are meant to fulfill missing information, such as the category of an object, its color or possible affordances. Being both researchers, Daniele and Anna are often busy and they do not want to be bothered too much (i.e., Roy is supposed to help them, not the contrary), even though they know that a little effort is required in order to help Roy in better performing the requested tasks. Hence, whenever such a collaboration is not too much invasive, it can be beneficial for everyone. However, Roy is an intelligent robot and when building this map, it pays attention to some details thanks to its capability of exploiting the **contextual environment**. In fact, it knows that when Daniele and Anna are approached too closely, they feel uncomfortable and not willing to help it (see Chapter 3). It also knows that Anna is usually more inclined to collaborate with it [44]. Hence, Roy keeps a certain distance before asking for help [118] and prefers to bother Anna instead of Daniele. Moreover, when Roy detects a new object, it is able to exploit the images taken with its RGB-D camera to make some inferences about the object. For example, as it has already seen some cups, it will not ask whether the new mug bought during a trip in Lapland is actually a mug (see Chapter 6). All this knowledge help Roy in minimizing both discomfort and effort required in helping it.

As Daniele is often focused on his research, he usually controls Roy through ambiguous commands [178]; however, Daniele is aware that Roy will be able to disambiguate such commands thanks to its capabilities of interpreting natural language by leveraging its representation of the environment. For example, Daniele loves drinking tea while working on a new paper in front of its laptop, but sometimes he misplaces his preferred mug. When Daniele asks Roy to *"take the mug next to the keyboard and fill that cup of tea"*, the robot might not know what to do. In fact, depending on whether the mug is already near the keyboard or not, the plan to be executed significantly changes. Whenever it is close to the mug, Roy just needs to grab it and pour some tea. Otherwise, the robot needs to locate the mug, bring it next to the laptop and, then, fill the cup with tea. Moreover, Daniele when referring to its mug, sometimes he uses the word *"mug"*, sometimes *"cup"* [74]. Fortunately, Roy is a cognitive robot and, by reasoning on the **contextual environment**, it will be able to interpret the command coherently with the status of the world it is operating into. Moreover, Roy is so intelligent that it knows that both *"mug"* and *"cup"* refer to the same entity (i.e., Daniele's beloved mug) (see Chapter 5). Moreover, as Anna loves to play her piano, the acoustic conditions of the environment do not allows Roy to properly recognize Daniele's commands. Hence, though Roy's speech recognition capabilities are able to produce a list of candidate transcriptions of Daniele's command, the correct one is not often ranked in first position. For example, the audio signal corresponding to the sentence *"take the mug next to the keyboard"* is often transcribed as the meaningless sentence *"deck the madness the keyboard"* and the correct transcription is lost somewhere within the hypothesis list. However, Roy is able to exploit its **contextual knowledge** to filter out transcriptions that are syntactically and semantically out of the specific application domain.

It is clear that, in order to successfully fulfill the assignments, Roy must be able to properly exploit the contextual information of its surrounding world. The goal of

this thesis is thus to contribute to the development of robotic platforms that display capabilities similar to those attributed to Roy.

1.2. Situated Interactions with Robots

Human-robot interactions are *situated*, in the sense that both the user and the robot are co-present in a shared physical world and make references to the environment, objects and entities populating it [36]. To this end, in physically situated settings, robots have to display diverse additional competencies. On the one hand, humans' actions are expected to be understood within the broader situational context; on the other hand, robotic platforms need to continuously mediate among their behavior and humans' one. Hence, for enabling robots to properly reason about their surroundings and coherently behave with it, it is essential to implement effective NL communications that take into account the operational context: due to limited perception capabilities, the robot's representation of the shared world represents a robust bridge between the physical world and the robot's reasoning capabilities. Thus, exploiting both the perceived and structured environmental **context** is an essential building block of any situated interactive system.

This section presents three situated scenarios addressed in this thesis. They all refer to the motivating example presented in Section 1.1, where the interplay between reasoning and context is essential, being a valuable mechanism for improving the robot's understanding and behaving capabilities.

1.2.1. Socially Acceptable Behaviors

The scenario depicted in Section 1.1 refers to a typical Symbiotic Autonomy situation (introduced in Section 2.5), where robots and humans help each other to go beyond their constraints and complete their tasks. In order to overcome their limitations, robots are allowed to ask for help. This is the case of Roy that needs Daniele and Anna's help for building the structured representation of the environment: as Roy is provided with limited perception capabilities, the only way it has to achieve the task of creating the map is to rely on humans' help. When the robot takes the initiative and asks humans for help, there is a change of perspective in the interaction, not yet specifically addressed by HRI studies. This specific HRI framework, where the robot exploits the humans' help, can become a widespread and practical approach. However, people are not always willing to help robots: even in human-human interactions, people have to pay attention to common social rules, in order to accept the other as a social collaborator. The same social rules are expected to be followed by robots, as they are meant to become social partners of our everyday lives in the next future.

It is thus essential, when designing the robot's behavior, to take into account the conditions under which the likelihood to obtain help is maximized. Such conditions are linked to the context in which the robot is operating. An effective robotic platform should be able to detect the contextual conditions under which it is allowed to establish a successful interaction with humans. Even more so, in the depicted scenario the interaction is essential for the robot itself, as the ultimate goal is obtaining information for fulfilling missing knowledge.

Before doing this, we need to assess: (i) the correspondence between Human-Human Interaction and HRI social rules, and (ii) the contextual factors that mostly impact on the establishment of the interaction itself. For example, Roy knows that Daniele and Anna do not like to be approached too closely. As a consequence, this factor is reflected in Roy's behavior that will keep a proper distance when approaching them [118]. Other constraints can be drawn from psychological studies, suggesting that, for example, collaborative attitude highly depends on the gender, with females being more collaborative than males [44, 122, 154, 158]. As a consequence, when asking for help, a robot might prefer females, as this decision would maximize the probability of receiving help.

One contribution of this thesis is represented by two user studies performed in such a scenario and presented in Chapter 3. The first user study [134] confirms the influence of conventional observable **contextual factors** (i.e., proxemics, gender, height) on people's collaboration attitude, while suggesting that the operational environment in which the interaction takes place (i.e., relaxing vs working) may not be significantly relevant. The second user study is carried out to better assess the influence of the activity performed by humans, when they are approached by the robot, as an additional and more compelling characterization of operational environment (i.e., standing vs sitting).

Hence, the overall findings of the above studies suggest that the attitude of users towards robots in the setting of Symbiotic Autonomy is indeed biased by contextual factors, which can thus be used to design socially acceptable robot behaviors.

1.2.2. Understanding Human Language

In the scenario depicted in Section 1.1, an important feature that Roy exhibits is the ability to react to spoken commands. This requires the understanding of the user utterance with an accuracy able to trigger the robot reaction. However, the correct interpretation of linguistic exchanges depends on manifold dimensions, including physical, cognitive and language-dependent aspects related to the environment.

When Roy is asked to *"take the mug next to the keyboard and fill that cup of tea"*, the robot has to process the audio signal corresponding to the uttered command, in order to feed its language understanding capabilities with the correct transcription. This is not a trivial issue in a real scenario, as environments are often noisy and spoken language is affected by misspelling, repetitions, and involuntary pauses.

A contribution of this thesis has been the design and development of a practical yet robust re-ranking approach for generic Automatic Speech Recognitions (ASRs). The proposed technique exploits **contextual information** provided by the application domain, in order to filter out implausible transcriptions and to promote candidates that are more likely in the targeted domain. Such a domain-dependent information has been encoded through a grammar, augmented with semantic attachments about actions and entities populating the environment, designed to model typical commands expressed in scenarios that are specific to service robotics. The outcomes obtained through an experimental evaluation show that the approach is able to effectively outperform the ASR baseline, obtained by selecting the first transcription suggested by the ASR.

Moreover, though the transcription has been correctly recognized, the robot must

be able to make some assumptions and resolve diverse linguistic ambiguities in order to accomplish the requested task. First, all the objects referred by the command must exist into the environment. In situated scenarios, interactions are highly linked to the context, as the communication usually makes references to the environment and entities composing it. Second, Roy needs a structured representation of the physical environment, in order to enable reasoning over the operational environment, e.g., the Semantic Map introduced in Section 2.4. Moreover, mechanisms to ground linguistic symbols to the entities of such a structured representation are essential for several reasons. On the one hand, Roy must be aware of which objects it has to physically operate on in order to accomplish the task. On the other hand, a complete knowledge of the environment can be helpful to resolve inherent ambiguities of the language. For example, the first action of the above command (*"take the mug next to the keyboard"*) might assume two different interpretations, depending on where the Prepositional Phrase (PP) *"next to the keyboard"* is attached, i.e., either to the noun *mug* or to the verb *take*.

However, as introduced before, the language is meant to reflect the context in which it is used: whenever the mug and the keyboard are close into the environment, the PP will be attached to *mug* and Roy will just need to pick up the mug; otherwise, Roy has to reach the mug (that is somewhere into the environment) and bring it next to the keyboard, as the PP is attached to the verb *take*. Hence, the operational context in which the interaction takes place can change the interpretation of the language; in a command interpretation task, for a robot to meet the user's desiderata, it is fundamental to properly exploit the perceived context.

Another contribution of this thesis (detailed in Chapter 5) is a comprehensive framework to systematically exploit **contextual knowledge** for grounding language in a command interpretation task. The proposed framework is thus able to produce interpretations consistent with Frame Semantics [53] that coherently mediate among the world (with all the entities composing it), the robotic platform (with all its inner representations and its capabilities) and the pure linguistic level triggered by a sentence. The approach has been realized in a Machine Learning (ML) setting, where contextual knowledge extracted from the Semantic Map is directly injected within the learning algorithm. In particular, the overall understanding problem has been decomposed into a cascade of sub-tasks, each solved through a dedicated Hidden Markov Support Vector Machine (SVM^{hmm}); contextual information is reflected into features of the learning machinery.

Several experimental evaluations proved that the integration of linguistic and perceptual knowledge actually improves the quality and robustness of the overall interpretation process. Being the proposed approach language-independent with respect to the adopted techniques, experiments showed also that such effectiveness holds even when the system is applied to different languages.

Moreover, the application of such an approach requires a corpus of training examples to generate the ML models. Hence, another contribution of this thesis has been the creation of a dataset of robotic commands, designed to be general enough for any robotic platform meant to operate in a domestic environment. The corpus, called Human-Robot Interaction Corpus (HuRIC), contains a total of 897 commands in two languages (i.e., English and Italian), providing linguistic annotations, interpretations in terms of semantic frames, audio files, and a portion of the Semantic Map justifying

the interpretation. The corpus has been made available to the community.

Finally, the proposed computational paradigm has been implemented in an off-the-shelf tool, adaptive spoken Language Understanding chain For Robots (LU4R), released to the research community. This system has been already used by several teams in international robotics competitions.

1.2.3. Instructing Robots through Natural Language

In the running example of Section 1.1, Daniele and Anna actively support Roy in acquiring the required knowledge about the new home, by providing natural language feedbacks about objects of the environment. In fact, one of the main steps towards the deployment of robotic platforms in real scenarios concerns their capability to reference objects and locations within the operational environment. Even though research on visual perception is pushing forward the performance of such systems, they still cannot be always considered reliable enough to be used without human validation. Moreover, a purely visual perception system is often not able to provide a complete semantic description of the entities populating the environment and its output is often limited to geometric information about the world. In addition, in real deployments, a robot might need to learn the idiosyncratic language used by different individuals, so that word meanings may need to be learned and adapted through interaction. Enabling robots to properly interact with users plays a key role in the effective deployment of robotic platforms in domestic environments: robots must be able to rely on interaction to improve their behavior and adaptively understand their operational world. For example, in the context of Symbiotic Autonomy (Section 2.5) interaction is essential to fulfill missing information. This is the case of Human Augmented Mapping (HAM) (see Section 2.5.1), where the aim is to build a representation of the environment by relying on the interaction with the user.

A proper and effective acquisition of semantic attributes of targeted entities through users' feedback is thus essential for the task accomplishment itself. Roy, in turn, should be able to access and leverage the acquired knowledge to improve the teaching process. In fact, these approaches are affected by tutoring costs, as the user becomes the main source of knowledge for the robot: such online incremental learning of semantic attributes can be tedious for the tutor, whenever the robot does not exploit the acquired information to improve the interaction experience and minimize tutoring cost. Hence, it would be desirable that, at some point, the robot could autonomously acquire new knowledge, and become more and more independent of the human, as the learning process proceeds.

A contribution of this thesis relates to a technique presented in [173] and detailed in Chapter 6 to acquire dialogue policies for robot teaching tasks, so that Roy will be able to minimize the tutoring cost. The proposed approach relies on a Dialogue Manager (DM) modeled as a multi-objective Markov Decision Process (MDP), where the optimization problem is solved through Reinforcement Learning (RL) (Appendix A.1.2). The DM interfaces with an online incremental visual classifier, based on a Load-Balancing Self-Organizing Incremental Neural Network (LB-SOINN) (see Appendix A.2.1), that allows to capture **contextual information**, encoded here as images of the targeted object. Through the interplay between the DM and

the visual classifier, Roy is thus allowed to efficiently query Daniele and Anna only whenever it really needs help.

Experiments conducted in a simulated scenario show the effectiveness of the proposed solution, suggesting that contextual information coming from the perception can be properly exploited to reduce human tutoring cost. Moreover, we proved that a policy trained on a small amount of data generalizes well towards larger datasets: the proposed online scheme, as well as the real-time nature of the processing, are suited for an extensive deployment in real scenarios.

1.3. Thesis Contributions

Summarizing the contents of the previous sections.

Chapter 3: The Role of Context in Robot Behavior Modeling

This chapter presents an analysis of observable contextual factors that might influence the collaborative behavior of people towards robotic platforms. In detail, this analysis has been made through user studies by:

- introducing a systematic metrics to quantitatively measure the Collaboration Attitude;
- identifying possibly influencing contextual factors;
- assessing the significance of the identified factors through the collected data;
- defining some simple guidelines for designing robots' behavior in the context of Symbiotic Autonomy.

Related Publications

- Francesco Riccio, Andrea Vanzo, Valeria Mirabella, Tiziana Catarci, Daniele Nardi (2016). Enabling Symbiotic Autonomy in Short-Term Interactions: A User Study. In Social Robotics - 8th International Conference, ICSR 2016, Kansas City, MO, USA, November 1-3, 2016, Proceedings, pp. 796–807, Kansas City, MO, USA.
- Roberto Capobianco, Guglielmo Gemignani, Luca Iocchi, Daniele Nardi, Francesco Riccio, Andrea Vanzo (2016). Contexts for Symbiotic Autonomy: Semantic Mapping, Task Teaching, and Social Robotics. In Symbiotic Cognitive Systems, Papers from the 2016 AAAI Workshop, Phoenix, Arizona, USA, February 13, 2016., pp. 733–736, Phoenix, Arizona, USA.

Chapter 4: The Role of Context in Speech Recognition

This chapter proposes a domain-specific re-ranking function for transcription lists produced by a generic ASR. In particular, the contributions of this research are:

- the definition of contextual evidence extracted from a grammar, designed to parse domain-dependent commands in NL, and
- a thorough experimental evaluation of the proposed method, that highlights the impact of domain-specific information with respect to the addressed task.

Related Publications

- Andrea Vanzo, Danilo Croce, Emanuele Bastianelli, Roberto Basili, Daniele Nardi (2016). Robust Spoken Language Understanding for House Service Robots. Polibits, 54, pp. 11–16.

Chapter 5: The Role of Context in Language Modeling

In this chapter, a framework for grounded language interpretation of robotic commands is presented and discussed. In detail, this contribution is structured as follows:

- the definition of a ML framework for the interpretation of robotic commands;
- the injection of contextual evidence into the learning process that allows to interpret the command coherently with the operational environment and to solve language inherent ambiguities at predicate level;
- a quantitatively analysis of the contribution provided by the contextual information;
- the development of a linguistic resource that collects examples of annotated commands, together with structured representations of the operational environments in which such commands might be uttered[1];
- the release of an off-the-shelf framework for the interpretation of spoken commands in an HRI context[2].

Related Publications

- Andrea Vanzo, Danilo Croce, Roberto Basili, Daniele Nardi (2017). LU4R: adaptive spoken language understanding for robots. Italian Journal of Computational Linguistics, 3(1), pp. 59–76.
- Andrea Vanzo, Danilo Croce, Roberto Basili, Daniele Nardi (2017). Structured Learning for Context-aware Spoken Language Understanding of Robotic Commands. In Proceedings of the First Workshop on Language Grounding for Robotics, Vancouver, Canada, August 3, 2017., pp. 25–34, Vancouver, Canada.
- Andrea Vanzo, Luca Iocchi, Daniele Nardi, Raphael Memmesheimer, Dietrich Paulus, Iryna Ivanovska, Gerhard K. Kraetzschmar (2017). Benchmarking Speech Understanding in Service Robotics. In Proceedings of the AIxIA Workshop on Artificial Intelligence and Robotics (AIRO@AIxIA), Bari, Italy, November 14, 2017., pp. 34-40, Bari, Italy.
- Daniele Evangelista, Wilson Umberto Villa, Marco Imperoli, Andrea Vanzo, Luca Iocchi, Daniele Nardi, Alberto Pretto (2017). Grounding natural language instructions in industrial robotics. In Proceedings of the IROS 2017 Workshop "Human-Robot Interaction in Collaborative Manufacturing Environments (HRI-CME), Vancouver, Canada, September 24, 2017.

[1] `http://sag.art.uniroma2.it/demo-software/huric/`
[2] `http://sag.art.uniroma2.it/lu4r.html`

- Andrea Vanzo, Danilo Croce, Giuseppe Castellucci, Roberto Basili, Daniele Nardi (2016). Spoken Language Understanding for Service Robotics in Italian. In AI*IA 2016: Advances in Artificial Intelligence - XVth International Conference of the Italian Association for Artificial Intelligence, Genova, Italy, November 29 - December 1, 2016, Proceedings, pp. 477–489, Genova, Italy.

- Andrea Vanzo, Danilo Croce, Roberto Basili, Daniele Nardi (2016). Context-aware Spoken Language Understanding for Human-Robot Interaction. In Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016), Napoli, Italy, December 5-7, 2016., pp. 308–313, Napoli, Italy.

- Emanuele Bastianelli, Danilo Croce, Andrea Vanzo, Roberto Basili, Daniele Nardi (2016). A Discriminative Approach to Grounded Spoken Language Understanding in Interactive Robotics. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016, pp. 2747–2753, New York, New York, USA.

Chapter 6: The Role of Context in Dialogue Modeling

This chapter presents an effortless conversational interaction model to properly control the dialogue flow in a robot teaching task. In particular, this contribution is structured along the following steps:

- the definition of a multi-objective RL framework for the acquisition of semantic attributes of objects populating the operating environment;

- the systematic exploitation of contextual visual information, encoded as images of the targeted object, through a comprehensive ML architecture that is able to provide guesses to the dialogue manager with the aim of minimizing the tutoring cost;

- the definition of a dedicated MDP, for controlling the reliability level of the visual classifier;

- a quantitative analysis of the impact of such contextual information in minimizing the tutoring cost.

Related Publications

- Andrea Vanzo, Jose L. Part, Yanchao Yu, Daniele Nardi, Oliver Lemon (2018). Incrementally Learning Semantic Attributes through Dialogue Interaction. In Proceedings of the 17th Conference on Autonomous Agents and MultiAgent Systems, pp. To appear.

- Andrea Vanzo, Danilo Croce, Emanuele Bastianelli, Guglielmo Gemignani, Roberto Basili, Daniele Nardi (2017). Dialogue with Robots to Support Symbiotic Autonomy. In Dialogues with Social Robots - Enablements, Analyses, and Evaluation, Seventh International Workshop on Spoken Dialogue Systems, IWSDS 2016, Saariselkä, Finland, January 13-16, 2016, pp. 331–342, Singapore.

1.4. Thesis Organization

This thesis is organized into 7 chapters as follows.

Chapter 2	Review of the literature on HRI, with a special emphasis on Spoken Human-Robot Interaction (SHRI). Moreover, as preliminary for the thesis, a detailed discussion of the concept of Semantic Map is reported, along with the introduction of the Symbiotic Autonomy paradigm.
Chapter 3	Definition of a formal quantitative model for evaluating the Collaboration Attitude in the context of Symbiotic Autonomy. Identification of possible influencing contextual factors. User studies on the role of such contextual factors in the maximization of Collaboration Attitude. Analysis of the insights of the users studies for their application in robot's behavior modeling.
Chapter 4	Definition of a re-ranking approach for a generic ASR that relies on domain-specific evidence to improve the accuracy of the speech recognition.
Chapter 5	Definition of a systematic ML framework for the interpretation of robotic commands. Development of corpus for training/evaluating the approach. Identification of contextual information that contributes in resolving persisting ambiguities of language. Feature modeling for including such environmental evidence into the ML processes. Experimental evaluation to validate the approach. Development of a complete tool for the interpretation of NL commands.
Chapter 6	Introduction of a RL framework for modeling the conversational interactions in a teaching task. Definition of a novel technique to exploit perceptual information with the aim of minimizing the tutoring cost. Experimental evaluation of the proposed solution and preliminary deployment on a real robot.
Chapter 7	Summary, conclusions and final remarks of the thesis. Open questions and future directions.
Appendix A	Technical discussion of the mathematical algorithms and resources used in this thesis.

Chapter 2

Spoken Human-Robot Interaction: Problems and Resources

The goal of the research in Human-Robot Interaction (HRI) is to realize robotic systems that exhibit a natural and effective interaction with users: robots should be provided with sensory systems able to understand and replicate human communication, such as speech, gestures, socially-acceptable behaviors, voice intonation, pragmatic interpretation, and any other non-verbal interaction. HRI is indeed a very extensive research area, that involves many different problems, communication modalities, and solutions. As a consequence, the literature is expanding rapidly, as confirmed by the excellent survey by Goodrich and Schultz [72] (even though no longer fully up to date).

This chapter provides an overview of the problems, resources, and applications involved in HRI, with a particular emphasis on interactions in Natural Language (NL). In fact, as already outlined by the scenario depicted in Section 1.1, this thesis focuses on robotic platforms that are able to intelligently interact with humans in social environments through NL and improve the interaction experience by properly leveraging the full stack of information provided by the contextual knowledge. Hence, this chapter contextualizes the thesis contributions by providing a brief overview of HRI (Section 2.1), along with the more specific research areas of *Spoken Human-Robot Interaction (SHRI)* (Section 2.2) and *Dialogue Management in HRI* (Section 2.3). A more fine-grained discussion of each contribution with respect to the specific literature is provided in each chapter. Then the concept of *Semantic Map* is discussed as a valuable source of information for enabling robots to effectively interpret human language (Section 2.4). Finally, the novel paradigm of *Symbiotic Autonomy* is introduced as motivating philosophy of the thesis (Section 2.5), together with one of its possible applications: Human Augmented Mapping (HAM) (Section 2.5.1).

2.1. Human-Robot Interaction

Following the running example of Section 1.1, this thesis is placed in the branch of HRI research devoted to human-robot social interaction scenarios, where the

robotic device provides entertainment, teaching, and assistance to people. In fact, in the depicted working scenario, Roy is expected to help Daniele and Anna in performing domestic tasks, by paying attention to humans' socially acceptable rules. The research in the area has been promoted by both companies and academic groups. Among others, Softbank is probably one of the most active company in producing commercial robots for social HRI. Even though some of their robots are used to play soccer [85], they are mainly meant to operate as interactive robotic platforms, engaging the users and providing a communication as more natural as possible [57]. Mattel has developed a new version of the famous Barbie doll (called *Hello Barbie*[1]) with speech and language recognition capabilities, with the aim of entertaining young girls and boys through long-term conversations. However, research is still the most interesting source of ideas for social HRI. An increasing number of robots are being developed as therapeutic companions for elderly and children, more and more often deployed into hospitals. Valuable examples are the works proposed in [49, 50]. Again, the use of robot interaction in education is promoted by several works (see, for example, [3, 31, 54, 56, 79, 102]), where robotic partners are deployed in children teaching activities. For the type of users involved, in the above domains, proper social behaviors are essential for the correct achievement of the desired task.

However, as emphasized in Section 1.2.1, in this context robots are seen as more than *just* intelligent systems; they are supposed to exhibit socially acceptable behaviors, thus aiming at becoming companions for the humans. Proper behaviors often depend on the **operational context** in which the robot is operating, composed of people, activities and physical features of the environment. A wide range of works investigated such problem, mostly taking the humans' perspective [55, 86, 118, 122, 124, 159, 177]. The cited works aimed at determining the *best setting* for a robot, that has to accomplish a task assigned by the user. A novelty pursued by this thesis is a change of perspective with respect to the problem: instead of evaluating the *passive behavior* of the users represented by their preferences in an HRI, this thesis aims at assessing a *proactive behavior* of humans when asked for help, given changes in the operational context. This different perspective allows to both minimize the level of discomfort and determine the best setting for robots that approach humans and ask for help. A detailed description of the contribution is provided in Chapter 3.

2.2. Spoken Human-Robot Interaction

The adoption of Natural Language (NL) in HRI is the leading thread of this thesis, as for its wide applicability in most of the HRI tasks. In fact, provided that the ultimate goal of HRI is the design of natural interfaces, a communication can be established through several modes, ranging from gestural, haptic and verbal interactions. However, humans usually communicate through NL, which, for several reasons, can be considered one of the most effective ways of interaction. Language is natural, in the sense that *speaking* is a capability we learn during our childhood, without any particular training, but just through the acquisition of examples. *Speaking* is thus an activity we are able to perform even ignoring the syntactic rules of the targeted language. For this reason, researchers from different backgrounds have tried to apply

[1] `http://hellobarbiefaq.mattel.com/`

NL communication to HRI: we call this research endeavor Spoken Human-Robot Interaction (SHRI).

Research has applied SHRI to deploy robotic systems in a wide variety of environments. For example, some speech-based techniques have been used in manipulators [189], and wheeled platforms [11, 97]. Moreover, some robots on the market support vocal interactions with users, such as the NAO Humanoid [73]. However, when NL is used to interact with robots, the interaction modality is called *situated*, in the sense that humans and robots interact by making references to a shared environment, though having different perceptions/representations of it. In this scenario, the contextual knowledge plays a key role even in language modeling. Initial studies on situated SHRI can be traced back to SHRDLU [179], a system able to process natural language instructions to perform manipulation actions in a virtual environment. Due to the nature of the task and the problems involved, taking into account different forms of contextual knowledge into the interpretation process is an essential feature that any SHRI system should provide. For example, in [24], the problem of understanding humans' language is tackled in an integrated fashion with the Automatic Speech Recognition (ASR), by augmenting recognition grammar rules with semantic attachments and producing the final interpretation depending on background knowledge. Combinatory Categorial Grammars (CCGs) are used in [96] to parse transcriptions of robotic commands obtained through ASR, for supporting an HAM task. The meaning is then grounded to the environment by relying on an ontology describing their operational world. A similar grounding approach is used in [88]. In this work, the authors use Conditional Random Fields (CRFs) to train specific semantic parsers and Semantic Maps provide anchors for linguistic symbols. In [33], the authors propose a system for grounded interpretation of NL navigation instructions, uttered in virtual indoor environments. The semantic parser is based on a combination of a Context-Free Grammar (CFG) and multiple Support Vector Machines (SVMs), while the context is here leveraged to acquire the mapping between meaning of NL instructions and plans from humans' demonstrations. The symbol grounding problem is addressed also in [161], where statistical graphical models are used to enable a mapping between words and syntactic parse structures with concrete objects, places, paths and events in the real world. Again, a Semantic Map is used to represent the contextual knowledge for grounding the linguistic symbols. Conversely, in [114], active perception through vision is used as contextual information to ground NL route instructions into robot executable commands.

The take-home message of the above brief literature analysis is that language modeling acquires a specific nature when applied to a situated scenario, such as the one presented in Section 1.2. In fact, linguistic interactions are context-aware in the sense that both user and robot access and make references to the environment. An effective robot for HRI must be able to ground the meaning (and its components) into the physical world as the interpretation is strongly interlaced with what is perceived [160]. This thesis makes two steps in this direction, by proposing: (i) a re-ranking approach for a generic ASR, and (ii) a Machine Learning (ML) framework for interpreting natural language coherently with the operating environment. The former is designed through a cost function that leverages **contextual knowledge** extracted from the application domain. Such a domain-dependent evidence allowed thus to improve the accuracy of a generic free-form ASR and to properly select the

most promising transcribed sentence. The latter is realized through the systematic injection of **contextual features** coming from the environment, directly into the learning/tagging process. Moreover, instead of relying on further interactions to cope with out-of-vocabulary words, the proposed approach makes large use of models of Distributional Semantics (DS), that improves the robustness of the overall system in terms of generalization capabilities. The combination of the proposed components allows to realize SHRIs systems that robustly recognize and interpret NL instructions with respect to the operational environment. The above contributions are reviewed in Chapters 4 and 5, respectively.

2.3. Dialogue Management in Human-Robot Interaction

An interaction is, by definition, an exchange of information between two or more components. In HRI, this exchange may assume different modalities and often such modalities are jointly modeled to achieve the desired goal ([26, 140, 145, 135]). However, when this process is performed by using NL as the main information carrier, the interaction becomes a dialogue or *dialogic interaction*. For this reason, Dialogue Management is another essential task that any SHRI system is expected to perform.

Dialogue Management in Robotics has been mostly applied in a task-based setting, in contrast with open-domain one. The difference is that while in the former the dialogue is used to accomplish a specific task, in the latter dialogue does not have any particular goal, and interaction proceeds without any objective, nor a pre-defined scheme. Task-based dialogic interactions have been adopted to clarify and resolve miscommunication ([70, 71, 111, 112]) or clarify persisting ambiguities of the language ([108, 162]).

Other schemes of dialogic interactions, closer to the one presented in the scenario of Section 1.1, are used in the context of the Symbiotic Autonomy paradigm, which this thesis refers to. In this case, dialogue is adopted to teach robots how to acquire knowledge about the operating environment [89, 95, 130, 187], or how to accomplish a given task, such as giving a tour [142], delivering objects [64], or manipulating them [66]. The underlying idea is that, by leveraging the humans' feedbacks through dialogic interaction, the robot will be able to fill the missing information.

In situated scenarios, some works investigated the problem of incrementally enhancing the interaction experience as the dialogue proceeds. This is often achieved by relying on the context provided by the dialogue history ([155]) or by the operational environment Garoufi and Koller [62]. More generally, in [89], the authors present a probabilistic approach to learn new referring expressions for robot primitives and physical locations in a map, by exploiting the dialogue with the user. The problem of Referring Expression Generation (REG) has also been tackled by Fang et al. [47] and further refined in [48]. They developed two collaborative models for REG. Both models generate multiple small expressions that lead to the target object with the goal of minimizing the collaborative effort. The problem of tackling the vocabulary in conversational systems has been addressed by [132]. They propose approaches that incorporate user language behavior, domain knowledge, and conversation context in

word acquisition, evaluating such methods in the context of situated dialogue in a virtual world.

This thesis proposes a dialogic framework that resembles all the above ideas and makes several contributions with respect to the presented literature. First, the interactive system is designed for the acquisition of semantic attributes of objects populating the environment and is able to cope with the situations depicted in the scenario presented in Section 1.1. In fact, in order to acquire visual information of the objects, the framework relies on an incremental visual classifier that does not need (i) to be trained up front, and (ii) a full specification of the objects composing the environment. These features allow exploiting the *contextual information* (i.e., RGB-D images of the targeted instances) to automatically recognize unseen objects and support a quick acquisition of the semantic map. Then, the proposed Dialogue Manager (DM) for the teaching task is entirely data-driven, enabling the deployment of the system in heterogeneous environments, and supporting interactions with people speaking different languages. Moreover, the dialogue policy can be acquired through a very small set of dialogue examples, enabling the deployment of this system in a long-term mapping scenario. This contribution will be discussed in Chapter 6.

2.4. Semantic Maps and Robot Perception

NL interactions take a specific nature when applied to Robotics: situated linguistic interactions are context-aware as both user and robot access and make references to the environment (e.g., perceived entities, their semantic properties, ...). Hence, a meaningful representation of the contextual environment is thus needed to connect the real world to the robot reasoning capabilities.

This section describes how to properly represent the contextual knowledge of the operational environment through *Semantic Maps*, required for supporting situated HRIs addressed in this thesis (Chapter 5). In fact, despite the great interest in designing formalisms and algorithms for building such representations, in this thesis this resource is used just as the source of information to enable linguistic inferences.

More formally, in line with [29], a Semantic Map is defined as the triple:

$$\mathcal{SM} = \langle \mathcal{R}, \mathcal{M}, \mathcal{P} \rangle \tag{2.1}$$

where:

- $\mathcal{R}$ is the global reference frame in which all the elements of the Semantic Map are expressed;
- $\mathcal{M}$ is a set of geometrical elements obtained as raw sensor data expressed in the reference frame $\mathcal{R}$ and describing spatial information in a mathematical form;
- $\mathcal{P}$ is the class hierarchy, a set of domain-dependent facts/predicates providing a semantically sound abstraction of the elements in $\mathcal{M}$.

$\mathcal{P}$ is modeled as a *(Monotonic) Inheritance Network*. It is worth emphasizing that we do not require that the knowledge acquired through perception is fully

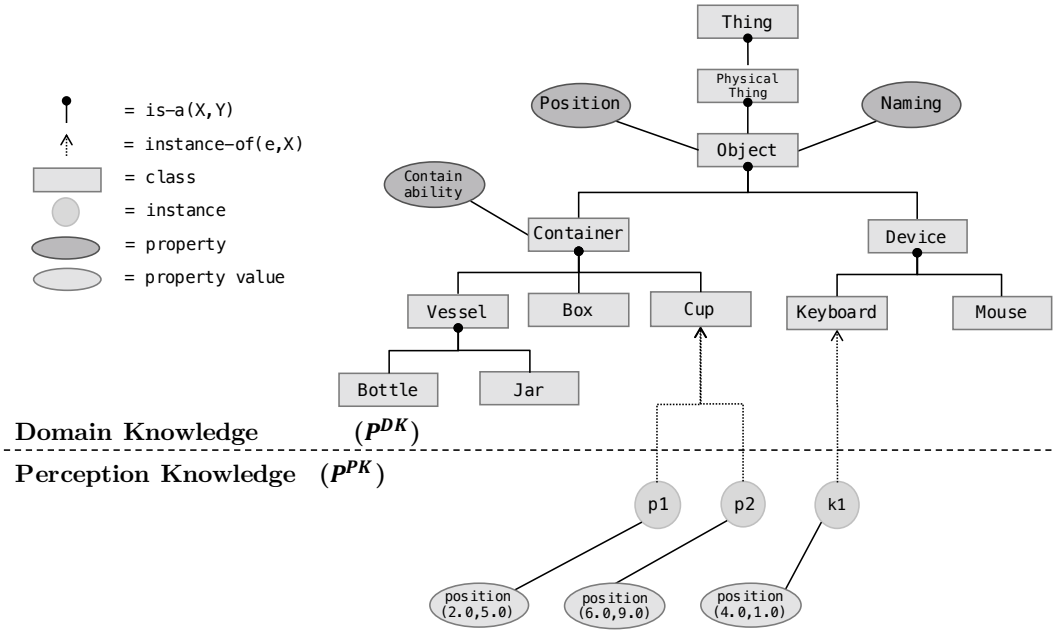


Figure 2.1. Sketch of the knowledge contained into a Semantic Map

consistent with the taxonomy of classes, as the Semantic Map is only used to support the linguistic processes addressed in this thesis. Hence, $\mathcal{P}$ is in turn decomposed into two layers:

$$\mathcal{P} = \langle \mathcal{P}^{\mathcal{DK}}, \mathcal{P}^{\mathcal{PK}} \rangle \tag{2.2}$$

where:

- the Domain Knowledge $\mathcal{P}^{\mathcal{DK}}$ is a conceptual knowledge base representing a hierarchy of classes, including their properties and relations, *a priori* asserted to be representative of any environment; it might be considered an intentional description of the robot's operation domain, while
- the Perception Knowledge $\mathcal{P}^{\mathcal{PK}}$ collects entities and properties specific to the targeted environment and represents the extensional knowledge, acquired by the robot.

The resulting structure of $\mathcal{P}$ is shown in Figure 2.1, highlighting both $\mathcal{P}^{\mathcal{DK}}$ and $\mathcal{P}^{\mathcal{PK}}$.

Domain Knowledge The *Domain Knowledge* provides the terminology (TBox) of the Semantic Map. It allows to define and structure the knowledge shared by different environments in the same domain. In particular, the Domain Knowledge proposed here (Figure 2.1, upper part) aims at modeling the hierarchy of classes, related to a domestic environment, and the domain-dependent semantic attributes.[2]

The hierarchy of classes of the $\mathcal{P}^{\mathcal{DK}}$ is modeled through `is-a`, e.g., `is-a(Cup, Container)`, and three specific properties: *Contain-ability*, *Naming* and *Position*.

[2]The attributes are assumed to be part of the Domain Knowledge $\mathcal{P}^{\mathcal{DK}}$.

Contain-ability defines that all the elements of a given class might potentially contain something. *Naming* provides a set of words used to refer to a class. Conversely, *Position* is a property that is instantiated only whenever there exists an entity of the targeted class. In fact, it determines the position of the entity within the grid map of the environment. The following predicates are used to characterize $\mathcal{P}^{\mathcal{DK}}$:

- `is-contain-able(C,t)` denotes that the Contain-ability property holds for all the objects of the class `C`, e.g., `is-contain-able(Cup,t)`;
- `naming(C,N)` defining `N` as the naming set, i.e., words that can be used to refer to the class `C`, e.g., `naming(Table,{table, desk})`.

The *Contain-able* property is modeled by relying upon a *Closed World Assumption*, so that whenever the property is not defined for a class, it is assumed to be false, e.g., `is-contain-able(Keyboard,f)`.

It is worth noting that, for each class `C`, its naming can be defined in different modalities: it can be acquired through dialogic interaction, by relying on the user's preferred naming convention, extracted automatically from lexical resources or defined a priori by a knowledge engineer. In this setting, alternative naming has been provided by the combined analysis of models of DS (see Appendix A.1.3) and Lexical Databases (e.g., WordNet), and validated by a knowledge engineer.

Perception Knowledge The Perception Knowledge (Figure 2.1, lower part) is the *ABox* of the Semantic Map. It represents the actual configuration of the current world. Hence, it is composed of elements that are actually present in the environment and perceived by the robot through its sensors.

$\mathcal{P}^{\mathcal{PK}}$ is defined through `instance-of(e,C)`, stating that the entity `e` is an entity of the class `C` and inherits all the properties associated to `C`. Moreover, whenever a new entity is included into the Semantic Map, its corresponding *Position* must be instantiated. To this end, `position(e,x,y)` represents the value of the *Position* property for a given entity `e` within the grid map, in terms of (`x`,`y`) coordinates. Moreover, on top of the Semantic Map, the function $d(\texttt{e1}, \texttt{e2})$ allows to return the Euclidean distance among the entities `e1` and `e2`. This value is essential to determine spatial properties of objects, e.g, whether two entities are far or near into the environment. For example, given two entities `instance-of(p1,Cup)` and `instance-of(k1,Keyboard)`, whose positions are `position(p1,2.0,5.0)` and `position(k1,4.0,1.0)` respectively, their Euclidean distance will be $d(\texttt{p1}, \texttt{k1}) = 4.47$.

The above formalization of the Semantic Map is essential to effectively shape the contextual information used within the linguistic processes addressed in Chapter 5.

2.5. The Symbiotic Autonomy Paradigm

Home environments constitute the main target location where to deploy robots, which are expected to help humans in completing their tasks. However, modern robots do not meet yet user's expectations in terms of both knowledge and skills. To this end, an increasing number of researchers are promoting HRI as a way to enable the robot to (i) understand the environment they are moving into and

(ii) accomplish tasks that would be otherwise unachievable. This field of research has been called *Symbiotic Autonomy*. Symbiotic Autonomy [139] or Symbiotic Robotics [38] is a general philosophy adopted for robot design. Under this principle, robots are not seen anymore as fully autonomous, solitary machines working in a static and unknown environment. Instead, they are seen as pervasive robotic systems working in symbiosis with people and their environments. Researchers have started to explicitly represent inside the robots their own limitations, in order to decide when to exploit human help to overcome their inabilities. Due to the nature of such an approach, Symbiotic Autonomy heavily relies on HRI for task execution. A straightforward example is when Roy asks Daniele and Anna's help in building the structured representation of the environment it needs in order to properly operate in its world. In fact, among others, HAM is an interesting and emerging task obeying such a paradigm, where the user helps the robot in building the Semantic Map. This task is detailed hereafter.

2.5.1. Human Augmented (Semantic) Mapping

In order to enable a robot to execute complex tasks and understand humans, environmental information needs to be semantically labeled. Semantic Mapping is the process of constructing the so-called Semantic Map (see Section 2.4), a synthetic representation of the environment that associates symbols to objects and locations of the world, along with semantic attachments useful to enable inferences on the targeted world. Once such a representation has been acquired, the robot will be able to execute commands like *"take the mug in the kitchen"*, without being tele-operated by the user and without the user's help in specifying the target position in terms of coordinates.

The problem of formalizing the *semantic knowledge* and generate semantic maps has been the focus of several works [77, 98].

A Semantic Map can be built by relying on hand-crafted ontologies and using traditional Artificial Intelligence (AI) reasoning techniques, unable to catch uncertainty inherently connected with semantic information coming from robot sensory system [60, 127]. The resulting map will be a static representation of the expert's perception of the world, preventing an effective adaptation to the end-user.

Other techniques [28, 121, 181] explore Semantic Mapping as a process where the purely automatic interpretation of perceptual outcomes is exploited to semantically enrich a geometric map. In this setting, no human effort is required and the process is performed completely autonomously. On the other hand, a detailed structure of the semantic properties is hard to acquire, some useful semantic information could be lost, and information cannot be gained through interaction.

Few approaches consider the human as part of the loop, by exploiting interactions in a human-robot collaboration setting [67, 151]; this process is often known as Human Augmented Mapping (HAM). In this case, the user is seen as an instructor (or tutor), that helps the robot (or learner) to acquire the required knowledge about the environment. In the range of HAM, it is clear that HRI acquires a special interest. In this framework, humans are seen as sources of information that the robot can interrogate to acquire novel knowledge. For example, the work by Zender et al. [187] proposes a system which is able to create conceptual representations of indoor

environments. They consider a robotic platform which owns a built-in knowledge. In this case, the user role is to support the robot in place labeling. Conversely, in [130], a multi-layered semantic mapping algorithm is presented. The algorithm combines information about the presence of objects and semantic properties related to space, such as room size, shape, and appearance. Whenever a user input is provided, it is combined as additional property about existing objects into the system. In the work by Nieto-Granda et al. [123], spatial regions are associated with semantic labels. The user is considered an instructor which helps the robot in selecting the correct labels.

More complex and advanced forms of human-robot collaboration are considered in a few works. The semantic map is built through a complete collaboration between human and robot. In fact, this interaction aims at objects recognition and positioning, rather than a simple place categorization and labeling. Such interactions are to some extent more complex and require advanced methods for natural language understanding. In fact, these systems are supposed to work even when non-expert and untrained users are considered. In this respect, multi-modal interaction represents an ideal communication means, as it able to deal with information of a different nature. For example, in [95] a system that aims at improving the mapping process by clarification dialogues between human and robot using natural language is introduced.

Following the view that considers the human operator as a fundamental source that the robot can query to acquire knowledge, Randelli et al. [133] introduce a system to generate semantic maps through multi-modal interactions. In this scenario, they use spoken languages to command the robot, and a vision system to enable the robot to perceive the objects that the user wants to identify and label. Gemignani et al. [67] generalize the approach to enable robots to incrementally build a semantic map of different environments while interacting with different users. Given a rich semantic map built with the help of the user, the system can perform qualitative spatial reasoning [65].

This thesis addresses a specific aspect of HAM, by focusing on the problem of enabling effective interactions during the semantic acquisition process. In fact, as soon as knowledge is being acquired, the robot should be able to exploit such information in order to make inferences that allow reducing the number of dialogue turns required to acquire the new knowledge. This contribution [173] will be discussed in Chapter 6.

Chapter 3

The Role of Context in Robot Behavior Modeling

In this chapter, the novel scenario brought by Symbiotic Autonomy (Section 2.5) is investigated, by addressing the **contextual factors** that may influence the interaction (Figure 3.1). In fact, given the current state of technology, autonomy significantly varies depending on the environment, on the task to be executed and on the robot platform itself. For example, as long as they do not feature any manipulator, robots will not be able to accomplish tasks such as grasping an object, opening a door or simply pushing a button. As already outlined in Section 1.2.1, to overcome such limitations, robots may ask for help, and accordingly, humans should be willing to help robots in achieving their tasks. This is the case of Roy that, in the scenario of Section 1.1, needs help to build the Semantic Map. The evaluation of how the robot should behave to successfully receive humans' help, and in which context it is better to ask for it, represents thus a novel scenario to be investigated.

This chapter reports and expands the results of a user study presented in [134], aiming at discovering and characterizing the influencing contextual factors of human attitude in helping a robot to accomplish its tasks. It is worth emphasizing that all the contextual factors taken into account are features of the environment that are observable and perceivable by the robot. The working hypothesis is thus that human attitude has not a constant value, but it depends on identifiable factors imposed by human physiology and by the context in which they are operating. The concept *Collaboration Attitude* and its systematic quantification are introduced to evaluate how the response of humans being asked by the robot for help is influenced by multiple contextual dimensions of the interaction and by what people are doing (i.e., ongoing activity). To this end, a first user study focused on the evaluation of the Collaboration Attitude [134], when specific factors such as *Proxemics* (i.e., relative pose of the interactive partners), *Gender* and *Height* of the experimenters and *Operational Environment* of the interactions are not constant. The data collected in this first experiment revealed that while Proxemics settings, Gender and (partially) Height play a key role in influencing the Collaboration Attitude of experimenters, the Operational Environment of the interaction, characterized by the location where the interaction takes place does not seem to be relevant. Hence, a second user study investigates the notion of *Activity* in which the human is involved, when the

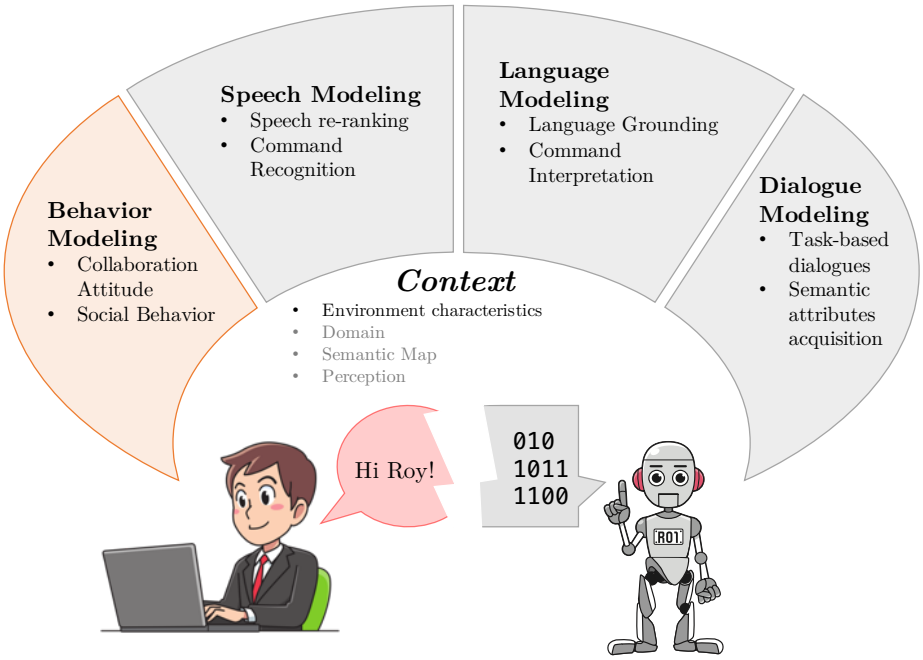


Figure 3.1. The behavior of the robot must be designed according to the environmental conditions.

interaction is triggered by the robot. To this end, rather than considering the context in which the interaction is carried out, we focused on studying the type of activity in which the user is involved, when (s)he is interrupted. This is a first step towards designing robots that are able to efficiently exploit their surroundings to improve their behavior.

The remainder of the chapter is organized as follows. Section 3.1 frames the work within the literature. Section 3.2 provides a formalization of Collaboration Attitude, defining our working hypotheses. Section 3.3 presents our system and the setup of the experiments. In Section 3.4, we report the experimental results of the first user study, while in Section 3.5 the empirical evidence of the second user study is presented. In Section 3.6, the results are discussed and, finally, in Section 3.7 we draw some conclusions and report the contributions of the chapter.

3.1. Related Work

Symbiotic Autonomy [139], or Symbiotic Robotics [38], describes a new paradigm in the collaboration between humans and robots, which is defined as a symbiosis between human and robot to enable a better coexistence of both. Several works in literature investigated how to enable such a cooperation among humans and robots. For instance, in [55] the authors study how to adapt robot behaviors to human preferences, while in [124] such a problem is faced by analyzing human responses to a robot offering domestic services. Differently from these works, where Collaboration Attitude is kept stable during the experiments, we assume that the Collaboration Attitude has not a constant value and depends on many factors such as general user

attitudes, human comfort and also on the type of activity that involves the user at the moment of the interaction.

Several user studies try to formalize a baseline to establish robot behaviors that guarantee a proper level of comfort during human-robot interactions. For example, in [86, 122] the authors find the best setting for enabling socially acceptable behaviors in handing-over objects and in properly gazing at the interactive partner. In [159], a user study is conducted to compare human-robot interactions, that involve users with a different personality, gender, height, and pet ownership. Similarly, in [118] the robot autonomously estimates the *comfort-level* of the operators, by comparing gaze orientation and physical distances in order to adapt its behaviors to specific participants; [177] represent the interactions as *fuzzy-rules* that can be updated online by an external operator. However, none of the aforementioned studies assumes the perspective of the robot taking the initiative towards the user. In fact, while in these studies the focus is to shape the robot behavior in response to a human request, here we aimed at enabling the robot to interrupt users in order to evaluate which are the behavioral and contextual features maximizing their Collaboration Attitude. More related to our study, [139] and [138] present a study about the behavior of a robot that needs help in the context of Symbiotic Autonomy. Even though in both scenarios the robot is also allowed to query humans, they do not analyze factors that may influence humans attitude to collaborate. Thus, no quantitative formalization and evaluation of the Collaboration Attitude are provided.

Summarizing, most of the works in literature aimed at determining the *best configuration* for a robot, that has to carry out a task assigned by a user. Under this perspective, the goal is to minimize the level of discomfort that can be caused. The main difference between this work and those reported in the literature lies in the premises of the task, that is here characterized by a robot asking for help and a human that is supposed to support it. In fact, the focus here is on evaluating the Collaboration Attitude of the subjects (*proactive behavior*), rather than their preferences during a Human-Robot Interaction (*passive behavior*) – as in [158]. Hence, from this different perspective, we want to both minimize the level of discomfort and determine the best setting for robots that approach humans and ask for help.

According to a thorough review of the literature, this research is the first presenting an analysis of the Collaboration Attitude and that studies its enabling factors with the goal to allow for more natural human-robot interactions.

3.2. A Study on Collaboration Attitude in Symbiotic Autonomy

In order to study the collaborative inclination of humans towards robots, a quantitative measure that captures the concept of Collaboration Attitude needs to be defined. To this end, the Collaboration Attitude has been modeled as an N-point Likert scale as follows:

Definition 3.1 (Collaboration Attitude)**.** *The Collaboration Attitude measures the attitude of humans towards the requests for help of the robot in a Symbiotic Autonomy framework. Formally, it is quantified according to metrics defined on a scale of N points, where N is the number of tasks that the human is requested to*

accomplish. Precisely, the Collaboration Attitude assumes values in $[0, \ldots, N-1]$*, where* 0 *represents the lowest level of collaboration, i.e., the human is not willing to help at all, while* $N-1$ *represents the highest one, i.e., the human is willing to help the robot in all the tasks.*

The working hypotheses of the two user studies are formalized in the following.

User Study 1: Proxemics, Gender and Context

In this user study, the robot asks people for help in different Operational Environments (namely, *Relaxing* and *Working*), with different Proxemics settings (namely, *Intimate*, *Personal* and *Social*), and balancing the experimenters on their Gender and Height. The analysis of such factors generates a model of interaction that defines: (i) whether they actually influence the Collaboration Attitude, and (ii) the values that maximize it. Eventually, such a model may be used to shape the proper social behavior of robots asking for help, depending on the current working context. Operationally, the following four hypotheses constitute the core of this user study.

Hypothesis 1. *Collaboration Attitude is subject to different Proxemics settings.*

It is well known that among humans and robots, Proxemics has a key role in the interaction. Therefore, experiments aim at highlighting the importance of respecting the personal space in social interactions, even in the case where the interactive partner is a robot. Specifically, the aim is to estimate whether different settings of Proxemics might vary the Collaboration Attitude that the human shows, ranging from an *intimate* distance to a *social* one.

Hypothesis 2. *Collaboration Attitude is subject to the gender of the human.*

Humans' physical and social characteristics affect how they behave in different situations. Gender is one of the major features to be considered. Such a factor is usually considered in Human-Robot Interaction (HRI) studies, as males and females show different responses to equal stimuli.

Hypothesis 3. *Collaboration Attitude is subject to the height of the human.*

Robot appearance constitutes a key factor to be investigated when studying humans response to robot behaviors. The intuition is that shorter people perceive the robot differently than taller people and their Collaboration Attitude varies depending on such a perception.

Hypothesis 4. *Collaboration Attitude is subject to different operational environment.*

The operational environment of the interaction plays a central part in social interactions. Humans behave differently, depending on where they are and the contexts they are in. Consequently, a robot needs to consider these social elements.

User Study 2: Human Activity

In this second user study, a robot interrupts people in order to ask for help, approaching people that are involved in different activities. Hence, the operational hypothesis is the following.

Hypothesis 5. *Collaboration Attitude is subject to different human activities.*

The activity in which the person is involved when the robot asks for help affects the level of collaboration towards the human. Specifically, users are meant to be in a *Standing* activity, if they stand at a location or are walking – for example, whenever they are going to a meeting or attending a class, or equivalently if they are having a coffee. Conversely, users are in the *Sitting* activity, instead, if they are sitting in the open areas, for example taking a break, having lunch or studying.

3.3. Method

The degree of Collaboration Attitude in changes to the dependent factors has been analyzed through two subsequent user studies. However, the subjects' selection policy, apparatus, procedure and questionnaire are essentially the same. In fact, for each user study, different runs of the same experiment have been executed by interrupting users in different activities and asking them to confirm the activity in order to discard outliers. This section introduces the subject population, the tools that supported the execution of the user study, the procedure of the experiment and the questionnaire.

3.3.1. Subjects

All the experiments have been conducted in the Department Computer, Control, and Management Engineering at Sapienza University of Rome, in different areas of the campus. For example, in the first user study, the *Relaxing* environment corresponds to the area in front of the vending machines, where people take a break from their activities, while in the *Working* environment the experiments took place in the corridor facing the offices. In the second user study, all the experiments have been performed in the department courtyard. Due to the nature of the user studies, the users have been randomly selected and approached, drawing from a set of students with homogeneous characteristics, all of them between 20 and 30 years old (i.e., 78 participants for User Study 1, 206 for User Study 2). Moreover, the experiment is completed in a *between group* design, so that every user participated only once and the data collected is not biased by repetitions of the experiments by the same user. Participants have not been compensated, nor have they provided any consensus for taking part in the experiment. This choice has been necessary to prevent possible bias in the results of the experiments.

3.3.2. Apparatus

In both user studies, the deployed robot is the same modified version of the TurtleBot Robot (see Figure 3.2). While the base remains unaltered, the structure on top of it has been customized, in order to make the robot taller with respect to the standard

Figure 3.2. Modified TurtleBot robot. The platform deployed is higher than the standard version, and features a tablet which is used to carry out interactions with users.

version. In fact, it is 98 cm high and it features a tablet on top as an interface for spoken interactions. We allow users to have *short-term dialogues* with the robot, to support the estimation of the attitude of the human to help the robot in performing its tasks. The short-term dialogue system is composed of two main modules: (i) an Automatic Speech Recognition (ASR), that processes the acoustic signal of the users' speech and generates a set of possible transcriptions; (ii) a Dialogue Manager (DM) that manages the dialogic interaction. The ASR module has been realized through the Google Speech APIs, available within the Android environment, in an ad-hoc mobile application. The app is also in charge of managing the questionnaire presented to the user at the end of the interaction, through a touch-based Graphical User Interface (GUI). The dialogue flow is managed through an Artificial Intelligence Markup Language (AIML) Knowledge Base (KB).

3.3.3. Procedure

We conducted our studies both in closed and open areas, where the heterogeneity of both environment and population gives the opportunity to collect data for each value of the considered factors. The whole experiment is conducted in a Wizard-of-Oz fashion [136] and includes a predefined set of four phases, namely *Approach*, *Dialogue*, *Questionnaire* and *Homing*. During the *Approach* phase, the robot approaches the user that is not aware of being involved in the study until the questionnaire is displayed. Given the purpose of the study, only this phase slightly differs depending on the factors and their value. In fact, once the next user is selected, the robot notifies its presence and seeks for help. The robot asks the experimenter to keep his/her position. Afterward, the robot approaches the user within one of the Proxemics

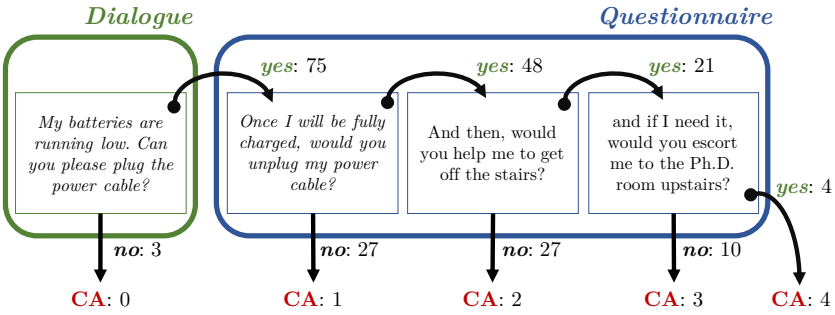


Figure 3.3. Questionnaire used in the first user study to evaluate the Collaboration Attitude, showing the numbers of users for the two choices (i.e., *yes* or *no*), at each stage of the questionnaire.

settings, without varying the orientation of the robot during the experiments, as other works [86, 164] focused on the relative orientation of the robot with respect to the user. Once the user attention is gained, the *Dialogue* phase is triggered and the robot asks to be helped in a particular task. After this short interaction, the robot displays the *Questionnaire* on the table aiming at completing the evaluation of Collaboration Attitude and collecting users' information. Once the questionnaire has been completely filled in, the *Homing* phase is executed, where the robot thanks the user and is guided towards its original position. It is worth emphasizing that the chosen characterizations of Operational Environment and Activity are done by taking into account the actual abilities of the robot perception.

3.3.4. Questionnaire

The data have been collected by asking the user to fill in a questionnaire that the robot displays on the tablet. The questionnaire is divided into two sections aiming at (i) quantifying the Collaboration Attitude, and (ii) collecting information about the user. Specifically, we characterize users by gathering information about gender, height, and acquaintance towards robotics. The Collaboration Attitude is mapped into a 5-point scale, measuring the number of positive responses of the experimenters to the robot requests, according to Definition 3.1. Hence, if we consider also the initial request (in the *Dialogue* phase), this variable takes values in $\{0, ..., 4\}$, where 0 is the case where the human is not willing to help the robot in any task and 4 the opposite situation. Figure 3.3 and Figure 3.4 show the requests posed to the experimenters. While the first request is part of the dialogic interaction, the remaining three are both uttered by the robot and displayed as part of the questionnaire. The numbers on each edge refer to the occurrences of a particular answer of both the user studies, i.e., *yes* or *no*. In particular, arcs labeled with *no* represent users giving up in helping the robot at a particular Collaboration Attitude request, while arcs labeled with *yes* count users that advanced through the different questions. For example, in the first user study (Figure 3.3), the 3 users neglecting the initial request achieved a Collaboration Attitude of 0, while in the second user study (Figure 3.4) the same level of Collaboration Attitude has been achieved by 31 users. Conversely, the users that satisfied all the robot requests and obtained a

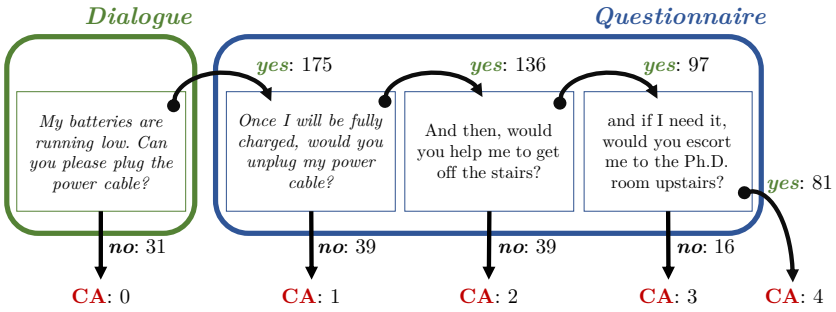


Figure 3.4. Questionnaire used in the second user study to evaluate the Collaboration Attitude, showing the numbers of users for the two choices (i.e., *yes* or *no*), at each stage of the questionnaire.

Collaboration Attitude score of 4 were 4 and 81, respectively. As one might expect, the engagement decreases as the requests become more and more demanding.

3.4. User Study 1: Experimental Results

This section reports the results obtained in the first user study. In Figure 3.5, means and standard errors of the Collaboration Attitude variable, obtained through a statistical analysis of the collected data, are plotted.

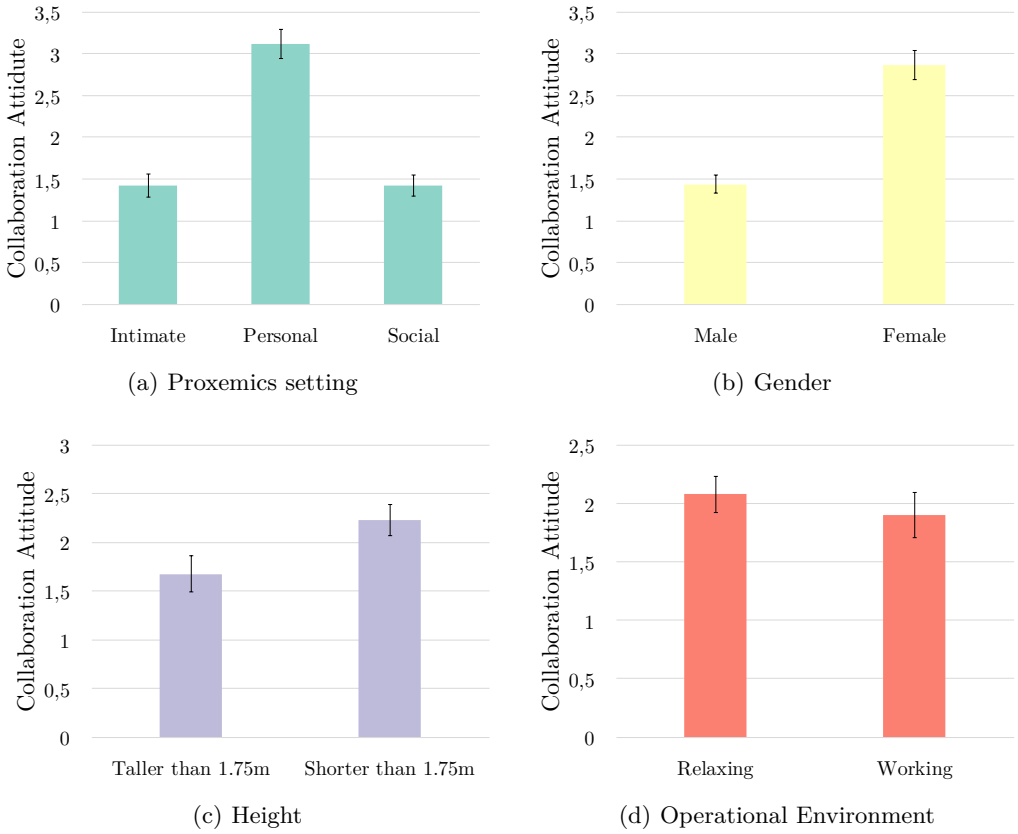
The Proxemics setting that maximizes the Collaboration Attitude is when the robot approaches the human with a Personal distance (Figure 3.5(a)). This result is in line with other user studies conducted in Human-Robot Proxemics [118, 122, 159], stating that humans' comfort is maximized within the Personal setting. The Intimate and Social distances give lower values of Collaboration Attitude.

When looking at the gender of the experimenters (Figure 3.5(b)), the mean of the Collaboration Attitude obtained by females is strikingly higher than the males' one. This represents a first indication that females are more inclined to help robots than males. The study of this factor is interesting, as it is known that males and females have different social behaviors.

Conversely, Figure 3.5(c) shows statistics of the Collaboration Attitude means to changes in height of the experimenters. The histograms suggest that shorter experimenters are more inclined to collaborating with respect to taller ones. However, this analysis might be influenced by several factors, such as the height of the robot and the different hight of male and female experimenters.

Despite the Relaxing Operational Environment seems to maximize the collaborative intentions of the experimenters (Figure 3.5(d)), the Collaboration Attitude is rather stable when different contexts are tested. As a consequence, the Operational Environment does not appear to be a perturbing contextual factor for the Collaboration Attitude.

In order to search for significant variations and test the operational hypotheses, we performed One-Way ANOVA over the different datasets. In Table 3.1, a sketch of the sample under consideration is shown. The populations of Proxemics and Operational Environment factors are completely balanced, with a population of 26



(a) Proxemics setting

(b) Gender

(c) Height

(d) Operational Environment

Figure 3.5. Collaboration Attitude means and standard errors of the first user study

Groups	*Count*	*Sum*	*Avg*	*Var*
Intimate	26	37	1.42	0.49
Personal	26	81	3.12	0.83
Social	26	37	1.42	0.41
Male	48	69	1.44	0.59
Female	30	86	2.87	0.95
Taller 1.75m	34	57	1.68	1.13
Shorter 1.75m	44	98	2.23	1.16
Relaxing	39	81	2.08	0.97
Working	39	74	1.9	1.46

Table 3.1. User Study 1: data statistics

Src of Var	*SS*	*df*	*MS*	*F*	*p-val*	*F crit*
			Proxemics			
Btw. Groups	49.64	2	24.82	42.95	$3.71 \cdot 10^{-13}$	3.12
Wtn. Groups	43.35	75	0.58			
Total	92.99	77				
			Gender			
Btw. Groups	37.71	1	37.71	51.84	$3.7 \cdot 10^{-13}$	3.967
Wtn. Groups	55.28	76	0.73			
Total	92.99	77				
			Height			
Btw. Groups	5.82	1	5.82	5.07	0.027	3.97
Wtn. Groups	87.17	76	1.15			
Total	92.99	77				
			Operational Environment			
Btw. Groups	0.63	1	0.63	0.52	0.47	3.97
Wtn. Groups	92.36	76	1.22			
Total	92.99	77				

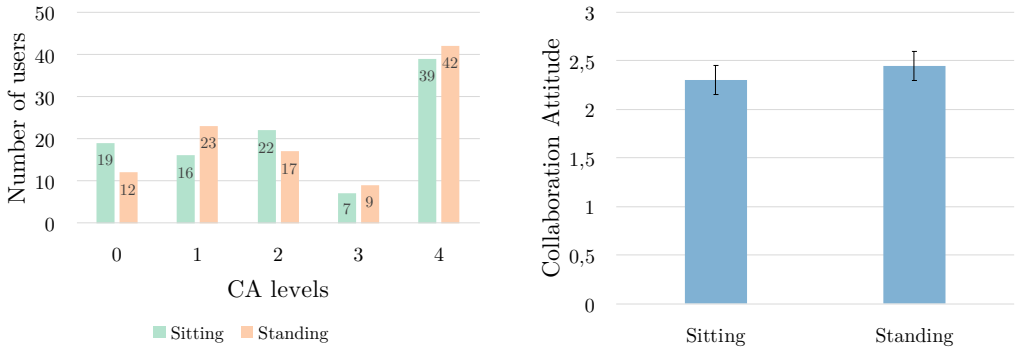
Table 3.2. User Study 1: One-Way ANOVA results

	Intimate vs. *Personal*	*Intimate* vs. *Social*	*Personal* vs *Social*
df	50	50	50
P(T<=t) two-tail	$9.6 \cdot 10^{-10}$	1	$4.1 \cdot 10^{-10}$

Table 3.3. t-Test: Two-Sample Assuming Equal Variances

elements for each Proxemics setting and 39 experimenters for both the Relaxing and Working groups. Conversely, the samples of the Gender factor are not balanced, with a majority of males with respect to females, i.e., 62% vs. 38% and a prevalence of shorter experimenters, i.e., 56% shorter vs. 44% taller.

Table 3.2 shows the ANOVA results by reporting the *P-value*, the *Sum of Squares* (*SS*), the *Degrees of Freedom* (*df*), the *Mean Squares* (*MS*), the *ratio of the two mean squares values* (*F*) and the *F critical value* (*F crit*). The Collaboration Attitude depends on the Proxemics setting chosen for the experiment (p-value < 0.05). In order to confirm the ANOVA results, we performed a post-hoc test through three t-tests, aimed at comparing each pair of groups. Table 3.3 shows the result of this additional analysis. As suggested by the means histogram, in the Personal distance humans act differently w.r.t. Intimate and Social settings (the *two-tailed p* values are lower than 0.05), whereas users seem to behave similarly in their Intimate and Social spaces. Also Gender and Height seem to be significant factor for the Collaboration Attitude. In fact, the One-Way ANOVA results allow to reject the null hypothesis in both cases (p-value < 0.05).



(a) Users' Collaboration Attitude levels divided with respect to the *Activity* factor

(b) Collaboration Attitude means and standard errors with respect to the *Activity* factor

Figure 3.6. Collaboration Attitude analysis of the second user study

Groups	*Count*	*Sum*	*Avg*	*Var*
Sitting	103	237	2.30	1.54
Standing	103	252	2.45	1.49

Table 3.4. User Study 2: data statistics

3.5. User Study 2: Experimental Results

In this section, the results of the second user study are reported. Again, the collected data have been analyzed through One-Way ANOVA test. Here the focus is on a more fine-grained discretization of the *Activity* that users are performing at the moment of the interaction.

Figure 3.6(a) shows the number of users grouped with respect to the Collaboration Attitude value that they achieve and divided according to the two values of the activity factor. Figure 3.6(b), instead, reports means and standard errors of the Collaboration Attitude. Interestingly, the plots show that *standing* users are slightly more inclined in collaborating with the robot, even though the statistical analysis (Table 3.5) confirms that the Collaboration Attitude values are firmly stable when different activities are compared. Thus, the Activity performed by the human does not appear to be a perturbing contextual factor. The populations for the two values of the activity factor are balanced as 103 users participated in each of the configurations.

Table 3.5 reports the ANOVA results by highlighting the *p-value*, the *ratio of*

Src of Var	*SS*	*df*	*MS*	*F*	*p-val*	*F crit*
			Activity			
Btw. Groups	1	1.09	1.09	0.47	0.49	3.89
Wtn. Groups	204	473.13	2.32			
Total	205	474.22				

Table 3.5. User Study 2: One-Way ANOVA results

the two mean squares values (F) and the *critical value* (F crit), the *sum of squares* (SS), the *degrees of freedom* (df) and the *mean square* (MS) for the given experiment. Instead, Table 3.4 reports for each of the considered settings, the number of users, the sum of Collaboration Attitude values, its average and variance.

3.6. Discussion

The aim of this investigation was to identify which contextual factors may influence the interaction between robot and human in the context of Symbiotic Autonomy, that is when the robot approaches the human to ask for help. The Collaboration Attitude has been introduced and defined and we set up two user studies to identify whether Proxemics, Gender, Height, Operational Environment and Activity have an influence on it. These factors have been chosen as observable features of the context in which the interaction takes place.

Not surprisingly, and in line with the findings of several works that address Proxemics as a key factor in human-robot interaction, Proxemics indeed plays a role. More specifically, this finding could be explained by two elements: the control that humans exercise in their Intimate space and the robot size. In fact, the presence of the robot seems to be not relevant, when the interaction takes place at longer distances. These results are particularly interesting in the framework of Symbiotic Autonomy: they suggest that a robot asking for help should approach the user in his personal space, as this distance seems to be the most comfortable for humans.

The results obtained over the Gender factor are supported by the work in [122]. In their user study, in fact, male users are more diffident and place themselves significantly further from the robot than females. These results are also confirmed in the work in [158]. Specifically, they report a considerable difference of the comfort level within the intimate area when varying the gender of the users. Their results support that males impose a dominant territory that the robot is violating if it is positioned in their intimate areas. In Human-Human interactions, manifold psychological studies address this particular behavior. For example, the difference in cooperating between males and females has been pointed out in [44], where this evaluation is made upon the well-known *Dictator Game*. A further confirmation is provided by [154], where the gender dimension is analyzed within an experimental study of team performance. In conclusion, the above results, obtained in the setting of Symbiotic Autonomy, report that female experimenters show more interest in exploring a new collaboration with a robotic partner. Therefore, the robot behavior could be leveraged by allowing the robot to seek for help first by female subjects.

The height of the experimenters is another interesting feature that deserves a better investigation. In fact, few works consider the height of the robot as a contextual *dependent variable* in their controlled studies [159, 177]. However, they do not state or highlight any empirical result on the influence of relative heights of the robot and users onto the interaction. Conversely, we noticed an interesting behavior, when classifying users by their heights. Such a categorization has been made by considering the average among the subjects' height of the population under analysis and the $1.75m$ value has been chosen as an unbiased discriminant factor. However, the outcomes of such analysis (Table 3.2) could be influenced by the females which are

usually shorter than males, and much more inclined to a human-robot collaboration. In fact, 70% of the female experimenters are shorter than $1.75m$, while the remaining 30% are taller than $1.75m$. Conversely, the male population is almost completely balanced. Hence, with the available data, we can clearly establish whether the height of the experimenters plays a key role in a human-robot collaboration. Hence, this particular aspect deserves an additional analysis, by increasing the variability and the size of the sample, as well as the height of the robot.

While this is an interesting confirmation that approaching space and users' physiology must be carefully considered as contextual factors in designing also symbiotic robots, the aim was also to understand whether what the user is doing is also relevant: in other words, whether it is worth trying to characterize the situations where it is more effective for the robot to ask humans for help. As in the first study, the focus was on the Operational Environment (Working vs. Relaxing), the initial hypothesis relied upon the intuition that humans in a relaxing context are more inclined to a collaborative behavior. However, the results of such an experiment showed that there are not statistically significant differences when changing the Operational Environment. This finding could be explained by a strong focus on the social interaction with other humans in a relaxing domain and it may suggest that robots are not yet considered *social* partners. This factor trades off the nature of the working context, where people are usually busy with their tasks.

As the first user study did not provide a clear answer to the question, we implemented a second user study trying to provide a better characterization of the situation in terms of the activity performed by the human (Standing vs Sitting). However, somewhat unexpectedly, the analysis of the data collected in the experiment indicates that humans do not show a different attitude depending upon the activity variable.

As a consequence, it seems that the analysis of the situation where the symbiotic interaction takes place does not have a significant impact on the design of robots' behavior when asking humans for help. It is worth remarking that *activities* have been determined in accordance with elements that are recognizable through robot perception capabilities. In other words, we identified activities that the robot will be able to detect through its own sensors. Such an outcome can have several justifications addressed in the following. First, the embodiment of the robot prevents the establishment of an interaction between the robot and the user at the emphatic level. Second, as a direct consequence of the previous remark, robots are not yet considered as social partners and their presence within the environment is still seen as a novelty factor. Finally, humans may consider the robot requests not plausible as they were expecting to act as operators commanding the robot. In fact, the way humans interact is strongly related to how the social partner is considered and perceived (i.e., which capabilities/tasks humans are expecting the robot is capable to perform and achieve.). This seems to strongly bias interactions and collaborations [40] and surely needs to be the subject of further investigation. However, by looking at these findings, it is clear that Collaboration Attitude needs to be better evaluated, including additional elements that will become available to the robot as its perception capabilities improve. Discovering new enabling factors for Collaboration Attitude will help increasing robot's chances to be considered a social partner when shaping social behaviors in everyday scenarios spanning from

guidance in museums to assistance in shopping malls. In fact, as soon as robots will operate more frequently in human-populated environments, symbiotic autonomy will play a key role in achieving a productive coexistence and thus, the collaboration will become essential.

3.7. Contributions

This chapter addressed the attitude of human subjects towards collaboration in the social perspective adopted by the so-called Symbiotic Autonomy. This specific HRI scenario, where robots ask humans for help, can become a widespread and practical approach, provided that robots exhibit proper social behaviors. Hence, the work presented in this chapter relates to a study on Collaboration Attitude, where we found out that Collaboration Attitude has not a constant value and depends on different enabling **contextual factors**. In particular, *Proxemics* and *Gender* seem to have a strong influence on the users' attitude towards collaborating, being the Personal space the area in which humans feel more comfortable towards the collaboration. Moreover, in line with several psychological studies, we found further confirmations that females are more inclined to collaborating with a robot. Conversely, the role of humans' *Height* needs a further and more accurate investigation in future research, as it might be also related to the size of the robot. On the contrary, the *Operational Environment* in which the interaction takes place do not seem to impact on the Collaboration Attitude. Hence, we decided to further analyze Collaboration Attitude with respect to the Activity users are performing during the interaction. To this end, two settings, being *Standing* the case in which users stand at a location or are walking and *Sitting* when users are sitting in open areas having lunch or studying. Experimental campaigns, through a robot deployed in real scenarios, show that users' attitude towards collaboration does not change depending on the activity they are performing. Hence, the overall study suggests that, when generating robot social interactions, the situation where the interaction takes place is less relevant than the general attitude towards the robot.

Hence, the contributions of this chapter are: (i) the introduction of a systematic metrics to quantitatively measure the Collaboration Attitude (Definition 3.1), (ii) the identification of possibly influencing contextual factors (namely, Proxemics, Gender, Height, Operational Environment and Activity), (iii) the setup of the user study, that can be reproduced to collect new data on the same or new contextual factors, (iv) the analysis of the identified factors through a user study

In conclusion, the findings of this chapter suggest that when designing social behaviors of a robot that operates in the Symbiotic Autonomy paradigm, some contextual factors deserve a special attention. In fact, they provide a valuable source of information to design human-acceptable behaviors. Though the study focused on a small set of contextual factors, when Roy needs help (Section 1.1) in building the Semantic Map, it should respect the social guidelines provided by this study, in order to establish an effective HRI and, consequently, maximize the probability of receiving help. The findings of this chapter are highly related to the ones provided in Chapter 6. In fact, it shows that the actively perceived environment can be exploited to improve the generation of Semantic Maps when the user plays the role of tutor.

Chapter 4

The Role of Context in Speech Recognition

This chapter presents a re-ranking approach to increase the robustness of an *off-the-shelf* free-form Automatic Speech Recognition (ASR) system in the context of understanding human language in Human-Robot Interaction (HRI) scenarios (see Section 1.2.2). The usefulness of Roy (Section 1.1) is directly dependent on its ability to perform the desired tasks. Whenever tasks are assigned through spoken commands, the understanding of the utterance passes through the correct recognition of the speech. The idea underlying the proposed approach is that, relying on **contextual knowledge** extracted from grammars designed over specific domains, it is possible to improve the accuracy of the adopted generic ASR (Figure 4.1). In fact, most of the existing off-the-shelf ASRs are based on very well-performing statistical methods [78], that enable their adoption in everyday scenarios. Nevertheless, these tools rely on general-purpose language models and false positives might be generated in specific scenarios. For example, they may be optimized to transcribe queries for a search engine, that are characterized by different linguistic constructions with respect to a command for a robot. However, it is reasonable to expect that domain-specific scenarios provide knowledge and specific information that can improve the performance of any off-the-shelf ASR. In this regard, several works proposed techniques where a hybrid combination of free-form ASRs and grammar-based ASRs is employed to improve the overall recognition accuracy. This thesis proposes to adopt a domain-specific grammar to improve the robustness of an ASR system by relying on a *scaling-down* strategy. First, some of the grammar constraints are relaxed, allowing the coverage of shallower linguistic information. Given a grammar, two lexicons are extracted to recognize (i) the vocabulary of in-domain robotic actions (ii) the vocabulary of the entities in the environment. For each lexicon, a specific *cost* is defined to be inversely proportional to its correctness. The transcriptions initially receive a cost that is inversely proportional to the rank provided by the ASR system and, each time one of them is recognized by the grammar or a lexicon, the corresponding cost decreases. The more promising transcription is the one minimizing the corresponding final cost. The final decision thus depends on the combination of all the costs so that, even when none of the transcriptions is recognized by the complete grammar, their rank still depends on the lexicons. In

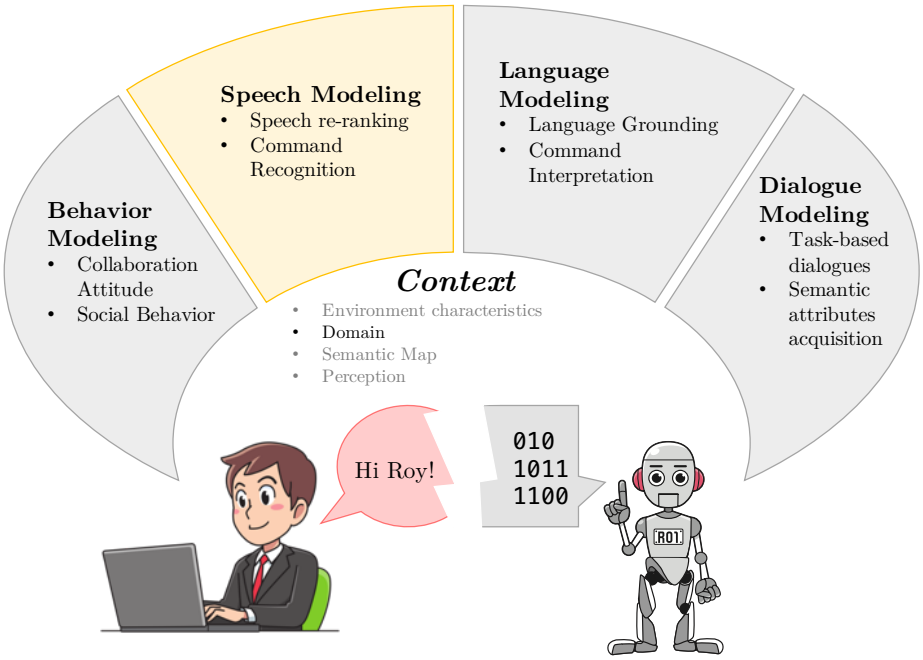


Figure 4.1. Speech recognition can be improved by taking into account domain-dependent information.

this way, those transcriptions that do not refer to any known actions and/or entities are accordingly penalized.

The proposed re-ranking strategy (introduced in [167]) has been evaluated on the Human-Robot Interaction Corpus (HuRIC) [13] a collection of utterances semantically annotated and paired with the corresponding audio file (HuRIC is part of the discussion of Chapter 5). This corpus is related with the adopted semantic grammar as this has been designed by starting from a subset of utterances contained in HuRIC. Experimental results show that the proposed method is effective in re-ranking the list of hypothesis of a state-of-the-art ASR system, especially on the subset of utterances whose transcriptions are not recognized by the grammar, i.e., no pruning strategy is applicable.

In the rest of the chapter, Section 4.1 provides an overview of the existing approaches to improve the quality of ASR systems. Section 4.2 presents the proposed approach and defines individual cost factors. In Section 4.3 an experimental evaluation of the re-ranking strategy is provided and discussed. Finally, Section 4.4 derives the conclusions and recaps the contributions.

4.1. Related Work

The robustness of ASR in domain-specific settings has been addressed in several works. In [119], the authors propose a joint model of the speech recognition process and language understanding task. Such a joint model results in a re-ranking framework that aims at modeling aspects of the two tasks at the same time. In particular, re-ranking of n-best list of speech hypotheses generated by one or more ASR engines

is performed by taking the Natural Language Understanding (NLU) interpretation of these hypotheses into account. On the contrary, the approach proposed in [15] aims at demonstrating that perceptual information can be beneficial even to improve the language understanding capabilities of robots. They formalize such information through Semantic Maps, that are supposed to synthesize the perception the robot has of the operational environment.

Regarding the combination of free-form ASR engines and grammar based systems, in [103] two different ASRs work together sequentially: the first is grammar-based and it is constrained by the rule definitions, while the second is a free-form ASR, that is not subject to any constraint. This approach focuses on the acceptance of the results of the first recognizer. In case of rejection, the second recognizer is activated. In order to improve the accuracy of such a decision, the authors propose an algorithm that augments the grammar of the first recognizer with valid paths through the language model of the second recognizer. In [42], a robust ASR for robotic application is proposed, aiming at exploiting a combination of a Finite State Grammar (FSG) and an n-gram based ASR to reduce false positive detections. In particular, a hypothesis produced by the FSG-based decoder is accepted if it matches some hypotheses within the n-best list of the n-gram based decoder. This approach is similar to the one proposed in [76], where a *multi-pass decoder* is proposed to overcome the limitations of single ASRs. The FSG is used to produce the most likely hypothesis. Then, the n-gram decoder produces an n-best list of transcriptions. Finally, if the best hypothesis of the FSG decoder matches with at least one transcription among the n-best, then the sentence is accepted. A hybrid language model is proposed in [105]. It is defined as a combination of an n-gram model, aiming at capturing local relations between words, and a category-based stochastic context-free grammar, where words are distributed into categories, aiming at representing the long-term relations between these categories. In [82], an interpretation grammar is employed to bootstrap Statistical Language Models (SLMs) for Dialogue Systems. In particular, this approach is used to generate SLM specific for a dialogue move. The models obtained in this way can then be used in different states of a dialogue, depending on some contextual constraints. In [104], n-grams and FSG are integrated in one decoding process for detecting sentences that can be generated by the FSG. They start from the assumption that sentences of interest are usually surrounded by carrier phrases. The n-gram is aimed at detecting those surrounding phrases and the FSG is activated in the decoding-process whenever start-words of the grammar are found.

All the above approaches can be considered complementary to the one proposed here. However, the advantages of the presented method are mainly in the simplicity of the proposed solution and the independence of the resulting work-flow from the adopted free-form ASR system: the aim is to define a simple yet applicable methodology that can be usable in every robot.

4.2. Re-Ranking Speech Hypotheses through Domain-dependent Knowledge

This section reports the proposed approach to select the most correct transcription among the results proposed by an ASR system. Such a technique relies on the

semantic grammar proposed in [16]. This grammar is modeled around the task of interpreting commands for robots expressed in Natural Language (NL) by encoding (i) the set of allowed actions that the robot can execute, (ii) the set of entities in the environment that should be considered by the robot and (iii) the set of syntactic and semantic phenomena that arise in the typical sentences of Service Robotics in domestic environment.

Hence, consider the example command provided in the scenario of Section 1.1 *"take the mug next to the keyboard"*. A free-form ASR might produce a rank of possible transcriptions such as

1. *deck the madness the keyboard*
2. *texmag nexo the keyboard*
3. *take the mug next to the keyboard*
4. *deck them all exo the keyboard*
5. *take them all next to the keyboard*

In this case, the correct transcription is ranked as third. In order to choose this sentence, a cost function is applied to the hypotheses based on (i) the adherence to the robot grammar, as it describes the typical commands for a robot, (ii) the recognition of action(s) applicable/known to the robot (as for *take*) and (iii) the recognition of entities, like nouns referring to objects recognized/known to the robot, e.g., *mug* or *keyboard*. The cost function decreases along with the constraints satisfied by the sentence, e.g., the second sentence satisfies (iii), but not (i) and (ii) (as *texmag* is not an action); as a consequence, it results into a higher cost with respect to the third transcription. Before discussing the cost function as a ASR ranking methodology, the grammatical framework used here is defined, in line with [16].

4.2.1. Grammar-based SLU for HRI

Robots based on speech recognition grammars usually rely on speech engines whose grammars are extended according to conceptual primitives, generally referring to known lexical theories such as Frame Semantics [53] (more details in Chapter 5). Early steps of understanding language in HRI are based on ASR modules that derive a parse tree encoding both syntactic and semantic information based on such theory. Parse trees are based on grammar rules activated during the recognition and augmented by an instantiation of the corresponding semantic frame, that corresponds to an action the robot can execute. Compiling the suitable robot command proceeds by visiting the tree and mapping recognized frames into the final command.

The applied recognition grammar jointly models syntactic and semantic phenomena that characterize the typical sentences of HRI applications in the context of Service Robotics. It encodes a set of imperative and descriptive commands in a verb-arguments structure. Each verb is retained as it directly evokes a frame, and each (syntactic) verb argument corresponds to a semantic argument. The lexicon of arguments is semantically characterized, as argument fillers are constrained by

one (or more) semantic types. For example, for the semantic argument THEME of the BRINGING frame, only the type TRANSPORTABLE_OBJECTS is allowed. As a consequence, a subset of words referring to things transportable by the robots, e.g., *can*, *mobile phone*, *bottle* is accepted. A subset of the grammar for the BRINGING frame, covering the sentence *"bring the book to the table"* is reported hereafter:

`Bringing` → `Target Theme Goal` | ...

`Target` → *bring* | *carry* | ...

`Theme` → *the* `Transportable_objects` | ...

`Transportable_objects` → *mug* | *book* | *bottle* | ...

`Goal` → ...

When building the domain-specific lexicons, we will distinguish between terminals denoting entities (such as *mug*, *book*, *bottle* that belong to the lexicon of TRANSPORTABLE_OBJECTS) from the lexicon of possible actions (such as *take*, *bring* or *move* characterizing the actions of the frame BRINGING) as they will give rise to different predicates augmented with grammatical constraints. Moreover, transcribed sentences covered by the grammar, i.e., belonging to the grammar language, are more likely to correspond to the intended command expressed by the user, and should be ranked first in the ASR output.

4.2.2. A Grammar-based Cost Model for Accurate ASR Ranking

A first interesting type of constraint is posed by the ASR system itself. In fact, the rank proposed by an ASR system is usually driven by a variety of linguistic knowledge in the ASR device. A basic notion of cost can be thus formulated ignoring the domain of the specific grammar.

Given a spoken utterance v, let $\mathcal{H}(v)$ be the corresponding list of hypotheses produced by the ASR. The size $|\mathcal{H}(v)| = N$ corresponds to the number of hypotheses. Each hypothesis $h \in \mathcal{H}(v)$ is a pair $\langle s, \omega(s) \rangle$, where s is the transcription of v, and $\omega(s)$ is a cost attached to s. Let $p(s)$ be its position in the ASR systems ranking. According to this cost function, the higher is $\omega(s)$, the lower the confidence in h being the correct transcription.

Since many off-the-shelf ASR systems do not provide the confidence score for each transcription, in order to provide a general solution, only the rank is taken into account. Let v be a spoken utterance and $\mathcal{H}(v)$ the corresponding list of transcriptions, then, $\forall s \in \mathcal{H}(v)$ the *ranking cost* ω_{rc} is defined as follows:

$$\omega_{rc}(s, \theta) = \frac{p(s) + \theta}{\sum_{s' \in \mathcal{H}(v)} p(s') + \theta N} \tag{4.1}$$

where $p(s)$ corresponds to the position $(1, \ldots, |H(v)|)$ of s in $\mathcal{H}(v)$. Here θ is a smoothing parameter that enables the tuning of the variability allowed to the final rank with respect to the initial rank proposed by the ASR system.

The overall cost assigned to a transcription s depends on the ASR ranking, as well as on the grammar. Let $s \in \mathcal{H}(v)$, let ω_i be a parametric cost depending on the grammar $\mathcal{G}$, the overall cost $\omega(s)$ can be defined as:

$$\omega(s) = log(\omega_{rc}(s, \theta)) + \sum_i log(\omega_i(s, \alpha_i)) \tag{4.2}$$

where the different ω_i capture different aspects of the grammar $\mathcal{G}$ with scores derived from the grammatical or lexical criteria. Higher values of ω_i correspond to stronger violations. Moreover, $\omega_{rc}(s, \theta)$ is the ranking cost as in Equation 4.1, while α_i is the parameter associated to each cost ω_i.

This approach investigates three possible cost factors, i.e., $i = 1, 2, 3$, to enforce information derived by different grammatical, i.e., domain-dependent, constraints as follows:

- $\omega_G(s, \alpha_G)$ is the *complete-grammar cost* that is minimal when the transcription belongs to the language generated by the grammar $\mathcal{G}$, and maximal otherwise;
- $\omega_A(s, \alpha_A)$ is the *actions-dependent cost* that is minimal when the transcription explicitly refers to actions the robot is able to perform, and maximal otherwise;
- $\omega_E(s, \alpha_E)$ is the *entities-dependent cost* that takes into account the entities targeted by the commands, and is minimal if they are referred into the transcription s and maximal otherwise.

These cost factors are detailed hereafter.

Complete-grammar cost. When dealing with the language understanding in robotics, we might be interested in restricting the user sentences to a set of possible commands. This is often realized by defining a grammar covering the linguistic phenomena we want to catch. Moreover, if the grammar is designed to embed also semantic information as in [16], higher level semantic constraints can be included into the definition of the grammar. For example, the BRINGING action can be applied only to TRANSPORTABLE_OBJECTS; as a consequence, a transcription such as *bring me the fridge* is discarded by the grammar if the *fridge* is not a TRANSPORTABLE_OBJECTS.

Let $\mathcal{G}$ be a grammar designed for parsing commands for a robot R. Let $L(\mathcal{G})$ be the language generated by the grammar, i.e., the set of all possible sentences that $\mathcal{G}$ can produce. Then, the *complete-grammar cost* ω_G is computed as

$$\omega_G(s, \alpha_G) = \begin{cases} \alpha_G & \text{if } s \in L(\mathcal{G}) \\ 1 & \text{otherwise} \end{cases} \tag{4.3}$$

where $\alpha_G \in (0, 1]$ is a weight that measures the strength of the violation and can be used to weight the impact of an *"out-of-grammar"* transcription. Notice that the weight α_G can be either set as a subjective confidence or tuned through a set of manually validated hypotheses. If α_G is set to 1, no grammatical constraint is applied and the complete grammar cost has no effect.

Action-dependent cost. Robot specifications enable the construction of the lexicon of potential actions A, hereafter called $\mathcal{L}_A$. Let A be the set of actions that a robot can perform, e.g., MOVE, GRASP, OPEN. For each action $a \in A$, a corresponding set of lexical entries can be used to linguistically refer to a: we will denote such a set as $\mathcal{L}(a) \subset \mathcal{L}_A$.

The *actions-dependent cost* ω_A for a transcription $s \in \mathcal{H}(v)$ is thus given by:

$$\omega_A(s, \alpha_A) = \prod_{\forall w \in s} \alpha_A(w) \tag{4.4}$$

where $\alpha_A(w)$ is defined as:

$$\alpha_A(w) = \begin{cases} \alpha_A & \exists a \in A \text{ such that } w \in \mathcal{L}(a) \\ 1 & \text{otherwise} \end{cases} \tag{4.5}$$

$\alpha_A \in (0, 1]$ is a weight that favors words corresponding to actions that are in the repertoire of the robot. The weight α_A can be either set as a subjective preference or tuned over a set of manually validated hypotheses. Note that if α_A is set to 1, no actions-dependent constraint is applied and the corresponding cost is not triggered.

Entity-dependent cost. Exploiting environment observations can be beneficial in interpreting commands. Notice that the objects of the robot's environment are more likely to be referred by correct transcriptions rather than by the wrong ones, as these are usually "out of scope". Let $\mathcal{G}$ be the grammar designed for commands. Given the set of terminals of $\mathcal{G}$, in the lexicon, $\mathcal{L}_G$ a specific set of terms is used to make (explicit) reference to objects of the environment. For each entity e (e.g., MOVABLE_OBJECTS such as *mug*, *books*, ..., or FURNITURES, such as *table* or *armchair*) the set of nouns used to refer to e in the language $L(\mathcal{G})$ is well defined, and it is denoted by $\mathcal{L}(e)$.

The *entities-dependent cost* ω_E for a transcription $s \in \mathcal{H}(v)$ is thus given by:

$$\omega_E(s, \alpha_E) = \prod_{\forall w \in s} \alpha_E(w) \tag{4.6}$$

where $\alpha_E(w)$ is defined as:

$$\alpha_E(w) = \begin{cases} \alpha_E & \exists \text{ entity } e \text{ such that } w \in \mathcal{L}(e) \\ 1 & \text{otherwise} \end{cases} \tag{4.7}$$

and $\alpha_E \in (0, 1]$ is a weight that favors words corresponding to entities the robot is able to recognize in the environment. The weight α_E can be either set as a subjective preference or tuned over a set of manually validated hypotheses. Also α_E, when set to 1, produces no entity dependent constraint and corresponds to a null impact on the final cost.

4.3. Experimental Evaluations

The grammar employed in these evaluations has been designed in [1], lately improved in [16], and its definition is compliant to the Speech Recognition Grammar Specification [80]. The grammar takes into account 17 frames, each of which is evoked by

an average of 2.6 lexical units (i.e., verbs). On average, for each frame, 27.9 syntactic patterns are defined. Entities are clustered in 28 categories, with an average amount of items per cluster of 11.2 elements. An Actions Lexicon $\mathcal{L}(a)$ containing 44 different verbs has is extracted from the grammar. The Entities Lexicon $\mathcal{L}(e)$ is composed of 216 and 97 single and compound words, respectively, with a total amount of 313 entities.

The dataset of the empirical evaluation is the HuRIC corpus (see Section 5.4), a collection of utterances annotated with semantic predicates and paired with the corresponding audio file. The HuRIC version used in these experiments is composed of three different datasets, that display an increasing level of complexity in relation with the grammar employed. The *Grammar Generated* dataset (GG) contains sentences that have been generated by the above speech recognition grammar. The *Speaky for Robot* dataset (S4R) has been collected during the *Speaky for Robots project*[1] and contains sentences for which the grammar has been designed so that the grammar is supposed to recognize a significant number of utterances. While the grammar is expected to cover all the sentences in the GG dataset, this may be not true for the S4R one, as some sentences are characterized by linguistic structures not considered in the grammar definition. The *Robocup* dataset (RC) has been collected during the 2013 edition of the *Robocup@Home* competition [180] and represents the most challenging section of the corpus, given its linguistic variability. In fact, even referring to the same house service robotics, it contains sentences not constrained by the grammar structure, as, during the acquisition process, speakers were allowed to say any kind of sentence related to the domain.

The experimental evaluation aimed at measuring the effectiveness of the proposed approach. To this end, the cost function $\omega(s)$ has been used in different settings. The α_i can be used to properly activate/deactivate the costs operating on specific evidence. In fact, if $\alpha_i = 1$, the corresponding cost is not triggered. However, whenever a cost is activated, its parameter has been estimated through 5-fold cross validation (with one fold for testing), as well as the θ smoothing parameter of the ranking cost ω_{rc}.

Performances have been measured in terms of Precision at 1 (P@1), that is the percentage of correctly transcribed sentences occupying the first position in the rank, and Word Error Rate (or WER). All audio files are analyzed through the official Google ASR APIs [32]. In order to reduce the evaluation bias to ASR errors, only those commands with an available solution within the 5 input candidates were retained for the experiments.

4.3.1. Experimental Results

Table 4.1 shows the mean and standard deviation of the P@1 and the WER across the 5 folds. The results have been obtained by testing our cost function on the aforementioned HuRIC corpus. The transcriptions have been gathered in January 2016. The sizes of the GG, S4R and RC datasets were of 100, 97 and 112 utterances, each paired with 5 transcriptions derived from the ASR system. The proposed approach has been compared, where hypotheses are re-ranked according to our cost function $\omega(s)$, against two different baselines. In the first baseline (*ASR BL*), the

[1] `http://www.dis.uniroma1.it/~labrococo/?q=node/3`

	GG		S4R		RC	
	P@1	WER	P@1	WER	P@1	WER
ASR BL	74.00 ±6.52	3.66	84.71 ±7.57	2.61	79.55 ±10.66	3.89
Greedy	**94.79** ±0.12	4.33	93.58 ±4.43	1.09	79.30 ±7.96	5.00
ω_G	90.00 ±3.54	1.13	**93.98** ±6.36	0.89	78.64 ±9.59	3.92
ω_A	80.00 ±7.07	2.22	82.71 ±10.02	2.85	82.27 ±10.21	3.65
ω_E	78.00 ±5.70	2.97	83.66 ±6.04	3.00	**83.18** ±11.32	3.19
$\omega_{G,A}$	90.00 ±3.54	1.13	92.93 ±6.63	1.06	80.45 ±11.54	3.79
$\omega_{E,G}$	90.00 ±3.54	1.13	**93.98** ±6.36	0.89	82.27 ±10.71	3.23
$\omega_{A,E}$	83.00 ±2.74	1.94	86.72 ±5.42	2.21	**83.18** ±10.85	3.71
$\omega_{G,A,E}$	90.00 ±3.54	1.13	92.93 ±6.63	1.06	82.27 ±12.07	3.75

Table 4.1. Results in terms of $P@1$ and WER

best hypothesis is selected by following the initial guess given by the ASR, i.e., the transcription ranked in the first position. The second baseline (*Greedy*) selects the first transcription, occurring within the list, that belongs to the language generated by the grammar. Conversely, the row ω_G refers to the cost function setting when α_A and α_E are set to 1, i.e., just the cost ω_G is actually triggered. In general, $\omega_{i,j,k}$ refers to the cost function when the costs ω_i, ω_j and ω_k are considered.

The *Greedy* approach seems to be effective when the sentences are more constrained by the grammar, i.e., it is likely that the correct transcription is recognized by the grammar. In fact, this approach is able to reach high scores of P@1 in both GG and S4R datasets, i.e., 94.79 and 93.58, respectively. Moreover, when the *complete-grammar cost* is triggered, i.e., ω_G, $\omega_{G,A}$ and $\omega_{G,A,E}$, we get comparable results, specially on the S4R dataset, with a relative increment of +10.94%. These observations do not apply for the RC dataset, where the structures and lexicon of the sentences are not constrained by the grammar. In fact, the *complete-grammar cost* does not seem to provide any actual improvement. Conversely, we observe a drop in performance when the full constrained grammar is employed, i.e., both *Greedy* and ω_G. On the other hand, when the *actions-dependent* and *entities-dependent costs* are considered, the best results are obtained. In particular, ω_E and $\omega_{A,E}$ are able to outperform both the *ASR BL* and the grammar constrained approaches. This behavior seems to depict a sort of *scaling-down* strategy: when the grammar does not fully cover the sentence, or it is not available, it is still possible to rely on simpler, but more effective, information. Nevertheless, even though it does not perform the best, the strategy where all costs are triggered, i.e., $\omega_{G,A,E}$, seems to be the most stable across different sentence complexity conditions.

Further experiments have been conducted on the transcription lists employed in [15]. These have been gathered by relying on the same ASR engine, but almost two years earlier (May 2014). Hence, a different amount of sentences are employed

	GG		S4R		RC	
	P@1	WER	P@1	WER	P@1	WER
ASR BL	84.18 ±11.53	2.04	85.48 ±6.80	4.61	78.75 ±8.39	5.15
Greedy	**94.00** ±5.48	2.36	**95.78** ±5.79	0.62	74.96 ±5.33	5.80
ω_G	92.00 ±8.37	0.74	92.60 ±5.48	2.09	80.00 ±8.15	4.82
ω_A	86.00 ±13.42	1.47	85.48 ±6.80	4.30	82.50 ±6.85	2.69
ω_E	84.18 ±11.53	2.04	82.40 ±6.41	3.37	83.75 ±3.42	3.57
$\omega_{G,A}$	92.00 ±8.37	0.74	92.88 ±7.05	1.41	82.50 ±5.23	2.66
$\omega_{G,E}$	92.00 ±8.37	0.74	92.60 ±5.48	2.09	82.50 ±5.23	2.98
$\omega_{A,E}$	86.00 ±13.42	1.47	83.94 ±7.84	3.32	**90.00** ±8.39	1.85
$\omega_{G,A,E}$	92.00 ±8.37	0.74	92.88 ±7.05	1.41	83.75 ±3.42	2.66

Table 4.2. Results in terms of $P@1$ and WER obtained over data used in [15]

in this experiment. In fact, the GG, S4R and RC datasets are composed of 51, 68 and 80 lists, respectively. The results are shown in Table 4.2. A similar trend is observed, with both *Greedy* and *complete-grammar cost* reaching the highest scores in GG and S4R datasets. Even though the results obtained on these corpora are still comparable with the ones presented in [15], the interesting behavior observed on the RC dataset represents the main substantial difference. Even on this dataset, the trend seems to be the same, with the $\omega_{A,E}$ outperforming any other approach with relative improvements in P@1 up to +20.06%. The trend of $\omega_{G,A,E}$ is confirmed here, making it the best solution as the most stable approach.

4.4. Contributions

This chapter presented a practical approach to increase the robustness of an off-the-shelf free-form ASR system meant to produce the transcription of robotic commands in the context of HRI. The approach relies on **contextual evidence** extracted from grammars designed over specific domains. In particular, a cost is assigned to each ASR transcription, that decreases along with the number of constraints satisfied by the sentence with respect to adopted grammar. Despite the simplicity of the proposed method, experimental results show that the proposed method allows to significantly improve a state-of-the-art ASR system over a dataset of spoken commands for robots.

Hence, the contributions of this chapter are: (i) the introduction of a re-ranking function to select the most promising transcription of commands uttered in domain-specific applications, among the ones produced by a generic ASR, (ii) the definition of contextual evidence extracted from a grammar, designed to parse NL commands, and (iii) experimental evaluations of the proposed re-ranking function, that highlight the impact of contextual information with respect to the addressed task.

It is worth emphasizing that a robust ASR is fundamental for any Spoken Human-Robot Interaction (SHRI) scenario. In fact, as already explained at the beginning of the chapter, feeding the NLU system with correct transcriptions is essential for the successful achievement of the robot's assigned tasks. Under this perspective, this chapter provides a simple and robust approach for improving the accuracy of any generic ASR adopted in situated interactions.

Chapter 5

The Role of Context in Language Modeling

This chapter focuses on the problem of interpreting robotic commands expressed through Natural Language (NL) in situated scenarios (see Section 1.2.2), so that the produced interpretation coherently mediates among the world, the robotic platform and the pure linguistic level triggered by a sentence (Figure 5.1). This feature is essential to properly design robotic platforms that are able to effectively meet the user's intent, by resolving diverse language inherent ambiguities, such as when the Prepositional Phrase (PP) attachment issue modifies both syntax and semantics of a sentence. Referring to the scenario of Section 1.1, this is a capability that Roy performs, when it is able to react to the command *"take the mug next to the keyboard"*, according to Daniele's intent and the environment.

In order to accomplish such goal, **contextual information** extracted from the structured representation of the environment is directly injected into the learning/tagging process, thus making the interpretation directly dependent on the environment. To this end, the interpretation process has been modeled as a cascade of data-driven processors, based on sequential classifiers. Each step of the cascade is handled through a Hidden Markov Support Vector Machine (SVM^{hmm}), where both linguistic and contextual features are injected into the models. The approach leverages models of Distributional Semantics (DS), to increase the generalization ability across lexical variations. The proposed approach allows thus to (i) learn the interpretation function from scratch, relying on a corpus of annotated commands, (ii) inject grounded information directly within the learning algorithm, integrating linguistic and contextual knowledge, and (iii) extend the features space as more specific and rich information is made available.

Experimental evaluations show that, when contextual knowledge is paired with linguistic evidence, the injection of these dimensions in the interpretation process is beneficial for the correct interpretation of the real user intent.

This chapter is structured as follows. In Section 5.1, the problem of grounding NL interpretations in robotic operational environments is discussed in the view of previous research and achievements in literature. Section 5.2 provides a description of the overall framework, addressing both resources and techniques used to accomplish the given task. Results obtained through several experimental evaluations are

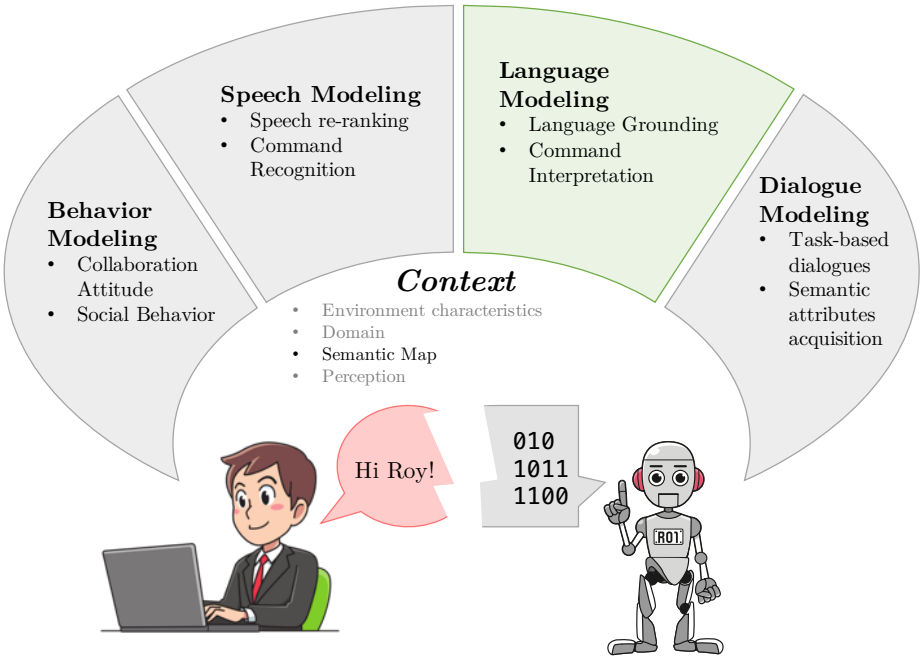


Figure 5.1. Operational context allows to ground human language to the environment.

reported in Section 5.3. In Section 5.4 the resource developed for training/testing the adopted Machine Learning (ML) techniques is presented, while Section 5.5 provides a description of the adaptive spoken Language Understanding chain For Robots (LU4R) framework, an off-the-shelf tool that implements the paradigms described in this chapter. Finally, Section 5.6 reports conclusions and contributions of the chapter.

5.1. Related Work

The approach proposed in this chapter makes use of grounded features extracted from a Semantic Map [126] modeling the entities in the environment, as well as *semantic* and *spatial* properties. Such features allow driving the interpretation process of the actions expressed by vocal commands. The realization of robots that are able to intelligently interact with users within human-populated environments requires techniques for linking language to actions and entities into the real world. Recently the research on this topic received an incredible interest (see, for example, the workshops on Language Grounding in Interactive Robotics [8, 144]).

Language grounding often requires the combination of the linguistic dimension and perception. For example, in [161], the authors make a joint use of linguistic and perceptual information. Their approach leverages active perception so that linguistic symbols are directly grounded in elements actively perceived. Again, in [106], a Natural Language Understanding (NLU) system called Lucia is presented, based on Embodied Construction Grammar within the Soar architecture. Grounding is performed using knowledge from the grammar itself, from the linguistic context, from the agent's perception, and from an ontology of long-term knowledge about

object categories and properties and actions the agent can perform. However, in the above works perceptual knowledge never modifies syntactic structures that can be generated by the parser when they are incorrect. Conversely, the system proposed here is able to deal with ambiguities at predicate level, allowing for selecting the interpretation that is mostly coherent with the operational environment.

Similarly to the framework presented in this thesis, the approaches in [84, 162] aim at grounding language to perception through structured robot world knowledge. In particular, in [162] the authors deal with the problem of out-of-vocabulary words, that are unknown to the robot, to refer to objects within the environment; the meaning of such words are then acquired through dialog. However, it is made use here of a mechanism based on models of DS [143, 115], while extracting grounded features through the lexical references contained in the Semantic Map. Hence, in this approach no further interactions are required, and the acquisition of synonymic expressions for referring to entities is automatically derived by reading large-scale document collections.

The problem of grounding semantic roles of a caption to specific areas of the corresponding video is addressed in [183]. Grounding is performed on both explicit and implicit roles. Semantic Role Labeling (SRL) follows a sequential tagging approach, implemented through Conditional Random Field (CRF). The problem is further stressed in [61], where Gao and colleagues studied a specific sub-category of the action verbs, namely the *result verbs*, that are meant to cause a change of state in the *patient* referred by the verb itself. In their framework, given a video and a caption, the aim is to ground different semantic roles of the verb to objects in the video, relying on the physical causality of verbs (i.e., physical changes that a verb may arouse within the environment) as features in a CRF model. Similarly, in [63] the problem of reasoning about an image and a verb is studied. In particular, the authors aimed at picking the correct sense of the verb that describes the action depicted into the image. In [23], the authors aim at resolving linguistic ambiguities of a sentence paired with a video by leveraging sequential labeling. The video paired with the sentence refers to one of the possible interpretations of the sentence itself. Even though they make a large use of perceptual information to solve a SRL problem, their system requires an active perception of the environment through RGB cameras. Hence, the robot must have the capabilities for observing the environment at the time the command is uttered. Again, in [4] the authors face the problem of teaching a robot manipulator how to execute natural language commands by demonstration, using video/caption pairs as a valuable source of information. Conversely, the proposed approach relies on a synthetic representation of the environment (see Section 2.4), acquired through active interaction [67]. It allows the robot to make inferences over the world it is working into, even though it is not actively and directly observing the surrounding environment. However, as the perception is injected in the interpretation process as features for the learning machine, the framework can be scaled to active perception, whenever vision information can be made available and encoded into features in real-time.

A different perspective has been addressed in [34], where the highly ambiguous problem of PP attachment of images' caption is resolved by leveraging the corresponding image. In particular, the authors propose a joint resolution of both semantic segmentation of the image and prepositional phrase attachment. In [90]

the authors exploit an RGB-D image and its caption to improve 3D semantic segmentation and co-reference resolution in the sentences. However, while the above works leverage visual context for the semantic segmentation of images or syntax disambiguation of captions, a synthetic representation of the context is used here to resolve semantic ambiguities of the human language, with respect to a situated interactive scenario. Hence, the proposed approach is able to cope with the correct semantics of an command that has been uttered in a specific context.

It is worth noting that approaches making joint use of language and perception have been proposed to model the language grounding problem also when the focus is on grounded attributes, as in [113, 93, 185]. Although the underlying idea of these works is similar to the technique presented here, our aim is to produce an interpretation at the predicate level, that can be in turn grounded in a robotic plan corresponding to the action expressed in an utterance. Therefore, the findings of such works can be considered complementary, as while they focus just on grounding linguistic symbols into entities and attributes, such a process is here leveraged for linking the whole interpretation to the current world.

To summarize, this work makes the following contributions with respect to the presented literature.

- The exploited perceptual information is extracted from a synthetic representation of the environment. This allows the robot to include information about entities that are not present in the same environment the robot is operating into.
- The discriminative nature of the proposed learning process allows to scale the feature space and to include other dimensions without re-structuring the overall system. Moreover, such property is used to evaluate the contributions provided by individual features.
- In this framework, perceptual knowledge is made essential to solve ambiguities at predicate level, thus affecting the syntactic interpretation of sentences according to dynamic properties of the operational environment.
- The system is robust towards lexical variation and out-of-vocabulary words and no interaction is required to solve possible lexical ambiguities. This is achieved through models of DS, used both as features for the tagging process and as principal component for grounding linguistic symbols to entities of the environment.
- Since the grounding function is a pre-processing, completely de-coupled step of the interpretation process, the mechanism is scalable to include further information that is not currently taken into account.

5.2. Grounded Interpretation of Situated Commands through Perceived Context

Human language is still one of the most natural vehicle of communication as for its expressiveness and flexibility: the ability of a robot to correctly interpret users'

commands is essential for proper Human-Robot Interaction (HRI). An effective communication in natural language between humans and robots is still challenging for the different cognitive abilities involved during the interaction. In fact, behind the simple command

"take the mug next to the keyboard" (5.1)

a number of implicit assumptions should be met in order to enable the robot for a successful execution of the command. First, the user refers to entities that must exist into the environment, such as the *mug* and the *keyboard*. Moreover, the robot needs a structured representation of the objects, as well as the ability to detect them. Finally, mechanisms to map lexical references to the objects must be available, in order to drive the interpretation process and the execution of a command.

This thesis argues that the interpretation of a command must produce a logic form through the integrated use of sentence semantics, accounting for linguistic and contextual constraints. In fact, without any contextual information, the command 5.1 is ambiguous with respect to both syntax and semantics due to the PP attachment ambiguity ([2, 35]). In the running example 5.1, the PP *"next to the keyboard"*, can be attached either to the Noun Phrase (NP) or the Verb Phrase (VP), thus generating the following different syntactic structures

[VP take [NP the mug [PP next to the keyboard]]] (5.2)

[VP take [NP the mug] [PP next to the keyboard]] (5.3)

that evoke different meanings as well. In fact, due to the high ambiguity of the *"take"* word, i.e., it can be noun or verb with different meanings [178], whenever the syntactic structure of the running command is 5.2, *"next to the keyboard"* refers to *"the mug"*. Hence, the semantics of the command evokes a ***Taking*** action, in which the robot has to take the mug that is placed next to the keyboard. Conversely, if the syntactic structure is 5.3, *"next to the keyboard‘"* is attached to the verb phrase, indicating that the mug is located elsewhere far from the keyboard. In this case, the interpretation of the command refers to a ***Bringing*** action, in which robot has to bring the mug next to the keyboard, that is the goal of the action.

It is clear that the structured representation of the environment is a discriminating factor for resolving syntactic/semantic ambiguities of language such as the PP attachment, as well as for providing the required knowledge in support of language grounding in a situated scenario.

In conclusion, this thesis fosters an approach for the interpretation of robotic spoken commands that is consistent with (i) the world (with all the entities composing it), (ii) the robotic platform (with all its inner representations and capabilities), and (iii) the linguistic information derived from the user's utterance.

5.2.1. Knowledge and Language for Robotic Grounded Command Interpretation

While traditional language understanding systems mostly rely on linguistic information contained in texts (i.e., derived only from transcribed words), their application

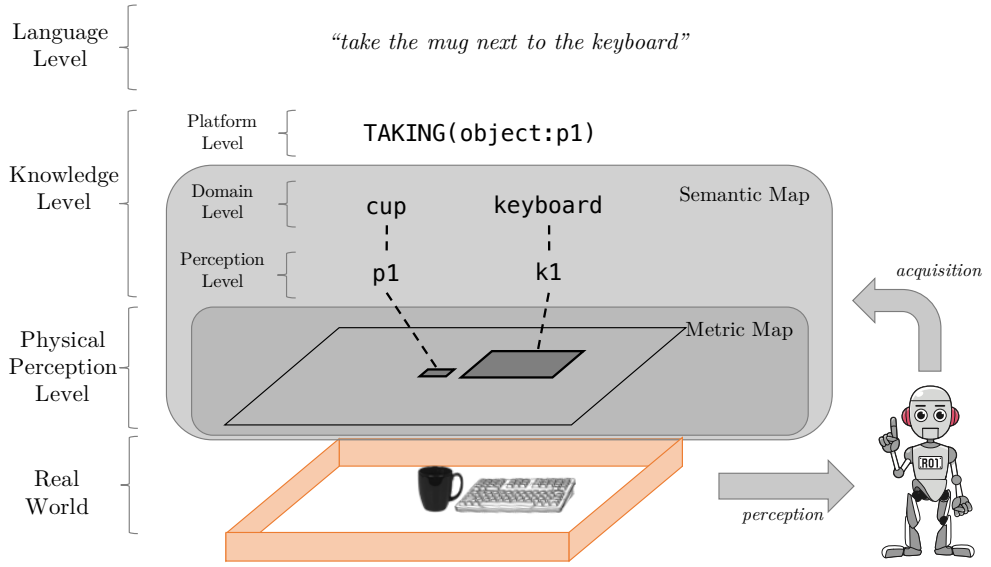


Figure 5.2. Layered representation of the knowledge involved in the interpretation of robotic commands

in HRI depends on a variety of other factors, including the perception of the environment. We might categorize these factors into a layered representation as shown in Figure 5.2. First, the *Language Level* is the governor of linguistic inferences: it includes observations (e.g., sequences of transcribed words), as well as the linguistic assumptions of the speaker; the language level is modeled through frame-like predicates. Similarly, evidence involved by the robot's perception of the world must be taken into account. The physical level, i.e., the *Real World*, is embodied into the *Physical Perception Level*: the robot is assumed to have a synthetic image of its world, where existence and possibly other properties of entities are represented. Such representation is built by mapping the direct input of robot sensors into geometrical representations, e.g., *Metric Map*. These provide a structure suitable for connecting to the *Knowledge Level*. Here *symbols*, encoded into the *Perception Level*, are used to refer to real world entities and their properties inside the *Domain Level*. The latter comprises active concepts the robot sees, realized in a specific environment, plus general knowledge it has about the domain. All these information play a crucial role during linguistic interactions, interplaying each other. The integration of metric information with notions from the knowledge level provides an augmented representation of the environment, called *Semantic Map* [126] (see Section 2.4). In this map, the existence of real world objects can be associated to *lexical* information, in the form of entity names given by a knowledge engineer or uttered by a user, as in Human Augmented Mapping (HAM) [41, 67]. It is worth noting that the robot itself is a special entity described at this knowledge level: it does know its constituent parts as well as its capabilities, that are the actions it is able to perform. To this end, an additional level is introduced (namely *Platform Level*), whose information is instantiated in a knowledge base called Platform Model ($\mathcal{PM}$). The main aim of such a knowledge base is to enumerate all the actions the robot is able to execute. While Spoken Language Understanding (SLU) for HRI have been mostly carried

out over the evidences specific to the linguistic level, e.g., in [33, 114, 12, 19], we assume that any convincing process should deal with all the aforementioned layers in an harmonized and coherent way. In fact, all linguistic primitives, including predicates and semantic arguments, correspond to perceptual counterparts, such as plans, robot's actions or entities involved in the underlying events.

In the following, the one of the building blocks of the proposed perceptually informed framework is introduced, defining the adopted interpretation formalism.

Frame-based Interpretation. A command interpretation system for a robotic platform must produce interpretations of user utterances. In this thesis, the understanding process is based on the theory of the Frame Semantics [53]; in this way, it is possible to give a linguistic and cognitive basis to the interpretations. In particular, the formalization promoted in the FrameNet [6] project is considered, where actions expressed in user utterances are modeled as *semantic frames*. Each frame represents a micro-theory about a real world situation, e.g., the actions of ***Bringing*** or ***Motion***. Such micro-theories encode all the relevant information needed for their correct interpretation, represented in FrameNet via the so-called *frame elements*, whose role is to specify the participating entities in a frame, e.g., the THEME frame element refers to the object that is taken in a ***Bringing*** action. Consider the running example 5.1 *"take the mug next to the keyboard"* provided in Section 5.2. Depending on which syntactic structure is triggered by the contextual environment, this sentence can be intended as a command whose effect is to instruct a robot that, in order to achieve the task, has to either

1. move towards a mug, and
2. pick it up,

or

1. move towards a mug,
2. pick it up,
3. navigate to the keyboard, and
4. release the mug next to the keyboard.

To this end, a language understanding cascade should produce its FrameNet-annotated version, that can be

$$[take]_{\textbf{\textit{Taking}}}\ [the\ mug\ next\ to\ the\ keyboard]_{\textsc{Theme}} \tag{5.4}$$

or

$$[take]_{\textbf{\textit{Bringing}}}\ [the\ mug]_{\textsc{Theme}}\ [next\ to\ the\ keyboard]_{\textsc{Goal}} \tag{5.5}$$

extracted coherently with the configuration of the environment.

In the following, the notation used for defining an interpretation in terms of semantic frames is introduced. It will be useful to support the formal description of the proposed framework. In this respect, given a sentence s as a sequence of words

w_i, i.e., $s = (w_1, ..., w_l)$, in this setting an interpretation $\mathcal{I}(s)$ in terms of semantic frames determines a conjunction of predicates as follows:

$$\mathcal{I}(s) = \bigwedge_{i=1}^{n} p^i \tag{5.6}$$

where n is the number of predicates evoked by the sentence. Each predicate p^i is in turn represented by the pair

$$p^i = \langle f^i, Arg^i \rangle \tag{5.7}$$

where:

- $f^i \in F$ is the label of the ith predicate evoked by the sentence, where F is the set of possible frames as defined in the $\mathcal{PM}$, e.g., ***Taking***, ***Bringing***, ..., and
- Arg^i is the set of arguments of the corresponding predicate p^i, e.g., [*the mug next to the keyboard*]$_{\text{Theme}}$ of the interpretation 5.4, while [*the mug*]$_{\text{Theme}}$ and [*next to the keyboard*]$_{\text{Goal}}$ for the interpretation 5.5.

Every $arg^i_j \in Arg^i$ is identified by a triple $\langle a^i_j, r^i_j, h^i_j \rangle$ describing:

- the argument span a^i_j defined as subsequences of s, so that the span $a^i_j = (w_m, \ldots, w_n)$ with $1 \leq m < n \leq l$, e.g., *"the mug next to the keyboard"* for 5.4 or *"the mug"* and *"next to the keyboard"* for 5.5;
- the role label $r^i_j \in R^i$ (or frame element) associated to the current span a^i_j and drawn from the vocabulary of frame elements R^i defined by FrameNet for the current frame f^i, e.g., the semantic roles Theme or Theme and Goal associated to the interpretations 5.4 and 5.5, respectively;
- the semantic head $h^i_j \in a^i_j$, as the meaning carrier word $w_k = h$ of the frame argument, with $m \leq k \leq n$, e.g., *"mug"* for the single argument of interpretation 5.4 or *"mug"* and *"keyboard"* for the arguments of interpretation 5.5.

Together with the arguments, Arg^i contains also the *lexical unit* (Lu) that anchors the predicate p_i to the text and is represented here through the same structure of arguments, e.g., the verb *take*. The two different interpretations of the running example 5.1 will be represented through the following structures

$$\begin{aligned} \mathcal{I}(s) \;=\; & \langle \textbf{\textit{Taking}}, \{ \\ & \langle (take), \text{Lu}, take \rangle, \\ & \langle (the, mug, next, to, the, keyboard), \text{Theme}, mug \rangle \} \rangle \end{aligned}$$

or

$$\begin{aligned} \mathcal{I}(s) \;=\; & \langle \textbf{\textit{Bringing}}, \{ \\ & \langle (take), \text{Lu}, take \rangle, \\ & \langle (the, mug), \text{Theme}, mug \rangle, \\ & \langle (next, to, the, keyboard), \text{Goal}, keyboard \rangle \} \rangle \end{aligned}$$

depending on the configuration of the environment.

In conclusion, semantic frames can thus provide a cognitively sound bridge between the actions expressed in the language and the execution of such actions in the robot world, in terms of plans and behaviors.

5.2.2. Grounding: a Side Effect of Linguistic Interpretation and Context

When interacting with a robot, users make references to the environment. This means that in order for the robot to execute the requested command s, the corresponding interpretation $\mathcal{I}(s)$ must be grounded: semantic frames provided by $\mathcal{I}(s)$ are supposed to trigger grounded command instances, that can be executed by the robot. Two steps are required for grounding an instantiated frame in $\mathcal{I}(s)$. First, the frame f^i corresponding to predicate $p^i = \langle f^i, Arg^i \rangle \in \mathcal{I}(s)$ must be mapped into a behavior. Then, all the frame arguments $arg^i_j \in Arg^i$ must be explicitly associated to their corresponding actors in the plan. In fact, role labels r^i_j are paired just with the argument spans a^i_j and semantic heads h^i_j corresponding to frame elements. However, a^i_j and h^i_j play the role of anchors for the grounding onto the map: each lexical item can be used to retrieve a corresponding entity in the environment. In this respect, let $\mathcal{E}_{\mathcal{PPK}}$ be the set of entities populating the Perception Knowledge $\mathcal{P}^{\mathcal{PK}}$ (defined in Section 2.4), collected as:

$$\mathcal{E}_{\mathcal{PPK}} = \{\mathtt{e} \mid \mathtt{instance\text{-}of}(\mathtt{e}, \cdot)\} \tag{5.8}$$

Then, for each entity $\mathtt{e}$, its corresponding naming can be gathered from the Domain Knowledge $\mathcal{P}^{\mathcal{DK}}$ as follows:

$$\mathcal{N}(\mathtt{e}) = \{w_\mathtt{e} \mid \mathtt{instance\text{-}of}(\mathtt{e}, \mathtt{C}) \wedge \mathtt{naming}(\mathtt{C}, \mathtt{N}) \wedge w_\mathtt{e} \in \mathtt{N}\} \tag{5.9}$$

that is: given the entity $\mathtt{e}$ and type $\mathtt{c}$, $\mathcal{N}(\mathtt{e})$ is composed of all the words in the naming set $\mathtt{N}$ associated to $\mathtt{c}$ that is defined into the $\mathcal{P}^{\mathcal{DK}}$ (see Section 2.4).

The proposed linguistic grounding function $\Gamma : arg^i_j \times \mathcal{P}^{\mathcal{PK}} \rightarrow \mathcal{G}_{arg^i_j}$ is carried out by estimating to what extent the argument arg^i_j matches the naming provided for the entities in $\mathcal{P}^{\mathcal{PK}}$. Hence, $\Gamma(arg^i_j, \mathcal{P}^{\mathcal{PK}})$ produces a set of entities $\mathcal{G}_{arg^i_j}$ maximizing the lexical distance between arg^i_j and $w_e \in \mathcal{N}(\mathtt{e})$, ordered depending on the real-valued lexical distance. Such lexical distance $g : h^i_j \times w_e \rightarrow \mathbb{R}$ is indeed estimated as a linear combination between word embeddings vectors of the semantic head h^i_j (associated to arg^i_j) and the words $w_\mathtt{e}$ [14]. Hence, the set of grounded entities $\mathcal{G}_{arg^i_j}$ can be defined as:

$$\Gamma(arg^i_j, \mathcal{P}^{\mathcal{PK}}) \rightarrow \mathcal{G}_{arg^i_j} = \{\mathtt{e} \in \mathcal{E}_{\mathcal{PPK}} \mid \exists w_\mathtt{e} \in \mathcal{N}(\mathtt{e}) \wedge g(h, w_\mathtt{e}) > \tau\} \tag{5.10}$$

where τ is an empirically estimated threshold obeying to application-specific criteria.

The lexical semantic vectors are acquired through corpus analysis, as in paradigms of DS (see Appendix A.1.3). They allow to control references to elements modeling synonymy or co-hyponymy, when arguments spans, such as *cup*, are used to refer to entities with different names, e.g., a `mug`. However, depending on how the function g is modeled, it is possible to inject non-linguistic features that might be meaningful for the grounding itself. In fact, at the moment only semantic head h^i_j and naming $w_\mathtt{e}$ are taken into account; hence, g neglects the contribution that, for example, adjectival modifiers may carry, e.g., the color of an entity can be helpful in disambiguating the grounded entity, whenever two entities, with different colors, of the same class are

present into the environment. The maximization of the similarity g between semantic head and entity naming corresponds to the minimization of the distance between the corresponding lexical semantic vectors and it can be extensively applied to grounding. Hence, g measures the confidence associated with individual groundings over the relevant lexical vectors. Although different settings of g (and therefore of Γ) can be designed [14], this mechanism is extensively used in this thesis to locate candidate grounded entities in the Semantic Map and to code them into perceptual features in the understanding process, hereafter described.

5.2.3. Contextually Informed Interpretation: the Language Understanding Cascade

The proposed interpretation framework is based on a cascade of statistical classification processes, modeled as sequence labeling tasks [17, 166, 170]. The classification is applied to the entire sentence and is modeled as the Markovian formulation of a structured Support Vector Machine (SVM) (i.e., SVM^{hmm} proposed in [5]). In general, this learning algorithm combines a local discriminative model, which estimates the individual observation probabilities of a sequence, with a global generative approach to retrieve the most likely sequence, i.e., tags that better explain the whole sequence.

In other words, given an input sequence $\mathbf{x} = (\vec{x}_1 \dots \vec{x}_l) \in \mathcal{X}$ of feature vectors $\vec{x}_1 \dots \vec{x}_l$, where $\mathbf{x}$ is a sentence and $x_i \in \mathbb{R}^n$ is a feature vector representing a word, the model predicts a tag sequence $\mathbf{y} = (y_1 \dots y_l) \in \mathcal{Y}^+$ after learning a linear discriminant function. Note that labels y_i are specifically designed for the interpretation $\mathcal{I}(s)$. In fact, this process is obtained through the cascade of the Frame Detection and Argument Labeling steps, where the latter is further decomposed in the Boundary Identification and Argument Classification sub-steps. Each of these is mapped into a different SVM^{hmm} sequence labeling task.

In the following, the ML approach is first introduced and then its application to each step of the cascade is addressed.

The Learning Machinery. The aim of a Markovian formulation of SVM is to make the classification of a word x_i dependent on the label assigned to the previous elements in a history of length k, i.e., $x_{i-k}, \dots, x_{i-1}$. Given this history, a sequence of k step-specific labels can be retrieved, in the form $y_{i-k}, \dots, y_{i-1}$. In order to make the classification of x_i dependent also from the history, the feature vector of x_i is augmented, by introducing a vector of transitions $\psi_{tr}(y_{i-k}, \dots, y_{i-1}) \in \mathbb{R}^l$: ψ_{tr} is a boolean vector where the dimensions corresponding to the k labels preceding the target element x_i are set to 1. A projection function $\phi(x_i)$ is defined to consider both the observations, i.e., ψ_{obs} and the transitions ψ_{tr} in a history of size k by concatenating the two representation as follows:

$$x_i^k = \phi(x_i; y_{i-k}, \dots, y_{i-1}) = \psi_{obs}(x_i) \;||\; \psi_{tr}(y_{i-k}, \dots, y_{i-1}) \tag{5.11}$$

with $x_i^k \in \mathbb{R}^{n+l}$ and $\psi_{obs}(x_i)$ does not interfere with the original feature space. Notice that the vector concatenation is here denoted by the symbol $||$, and that linear kernel functions are applied to different types of features, ranging from linguistic to world-specific features.

The feature space operated by ψ_{obs} is defined by linear combinations of kernels to integrate independent properties. In fact, through the application of linear kernels, the space defined by the linear combination is equivalent to the space obtained by juxtaposing the vectors on which each kernel operates. More formally, assuming that K is a linear kernel, e.g., the inner product, and being x_i, x_j two instances, each composed by two vector representations a and b (i.e., $x_{i_a}, x_{i_b}, x_{j_a}, x_{j_b}$), then the resulting Kernel $K(x_i, x_j)$ will be the combination of the contributions given by Kernels working on the two representations (i.e., $K_a(x_{i_a}, x_{j_a})$ and $K_b(x_{i_b}, x_{j_b})$, respectively), that can be approximated through the concatenation of vectors $x_{i_a}||x_{i_b}$ and $x_{j_a}||x_{j_b}$:

$$K(x_i, x_j) = K_a(x_{i_a}, x_{j_a}) + K_b(x_{i_b}, x_{j_b}) = \langle x_{i_a}||x_{i_b}, x_{j_a}||x_{j_b}\rangle \tag{5.12}$$

Conversely, $\psi_{obs}(x_i) = x_{i_a}||x_{i_b}$.[1]

At training time, the SVM learning algorithm LibLinear is used [46], and implemented in Kernel-based Learning Platform (KeLP) [52] in a One-Vs-All schema over the feature space derived by ϕ, so that for each y_j a linear classifier $f_j(x_i^k) = w_j\phi(x_i; y_{i-k}, \ldots, y_{i-1}) + b_j$ is learned. The ϕ function is computed for each element x_i by exploiting the gold label sequences. At classification time, all possible sequences $\mathbf{y} \in \mathcal{Y}^+$ should be considered in order to determine the best labeling $\hat{\mathbf{y}} = F(\mathbf{x}, k)$, where k is the size of the history used to enrich x_i, that is:

$$\begin{aligned}\hat{\mathbf{y}} \quad = F(\mathbf{x}, k) \quad &= \arg\max_{\mathbf{y}\in\mathcal{Y}^+}\{\sum_{i=1\ldots m} f_j(x_i^k)\} \\ &= \arg\max_{\mathbf{y}\in\mathcal{Y}^+}\{\sum_{i=1\ldots m} w_j\phi(x_i; y_{i-k}, \ldots, y_{i-1}) + b_j\}\end{aligned}$$

In order to reduce the computational cost, a *Viterbi-like decoding algorithm* (Figure 5.3) is adopted[2] to derive the sequence, and thus build the augmented feature vectors through the ϕ function. More details about the SVM^{hmm} mathematical framework provided, see Appendix A.1.1.

In the following, the different steps of the processing cascade are addressed individually.

Frame Detection. The processing cascade starts with the **Frame Detection (FD)** step, whose aim is to find all the frames evoked by the sentence s. It corresponds to the process of filling the elements p^i in $\mathcal{I}(s)$, and can be represented as a function $f_{FD}(s, PM, \mathcal{P}^{\mathcal{PK}})$, where s is the sentence, $\mathcal{PM}$ is the Platform Model and $\mathcal{P}^{\mathcal{PK}}$ is the Perception Knowledge. Assuming $s =$"*take the mug next to the keyboard*", then

$$\begin{aligned}f_{FD}(s, PM, \mathcal{P}^{\mathcal{PK}}) \quad = p^1 = \quad &\langle \textbf{\textit{Taking}}, \{ \\ &\langle\langle take\rangle, \textsc{Lu}, take\rangle, \\ &\ldots\}\rangle\end{aligned}$$

[1] Before concatenating, each vector composing the observation of an instance, i.e., $\psi_{obs}(x_i)$, is normalized to have unitary norm, so that each representation equally contributes to the overall kernel estimation.

[2] When applying $f_j(x_i^k)$ the classification scores are normalized through a softmax function and probability scores are derived.

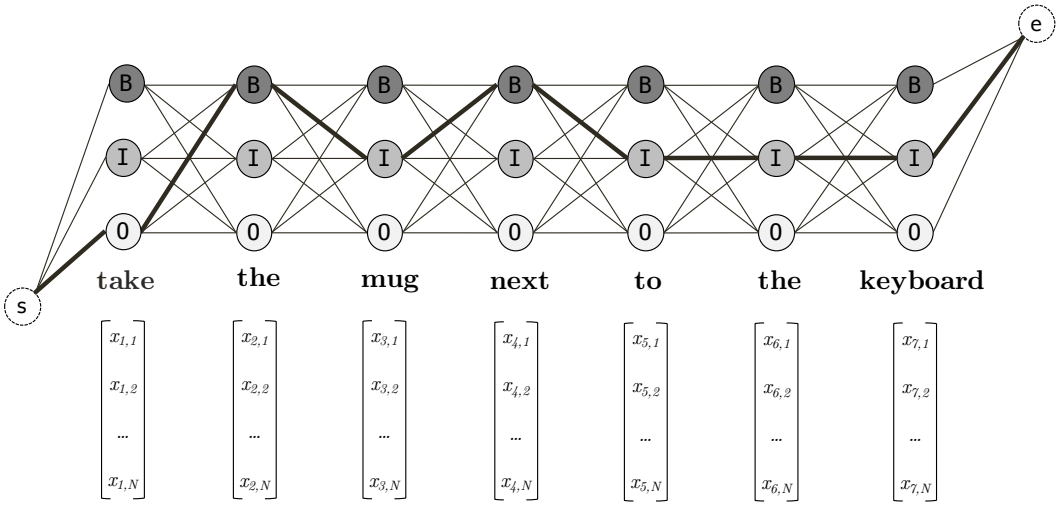


Figure 5.3. Viterbi decoding trellis of the Boundary Identification step (Section 5.2.3), for the running command *"take the mug next to the keyboard"*, when the interpretation 5.5 is evoked. The label set refers to the IOB2 scheme, so that $y_i \in \{B, I, O\}$. Feature vectors x_i are obtained through the ϕ function. The best labeling $\mathbf{y} = (O, B, I, B, I, I, I) \in \mathcal{Y}^+$ is determined as the sequence maximizing the cumulative probability of individual predictions.

for interpretation 5.4, while

$$f_{FD}(s, PM, \mathcal{P}^{\mathcal{PK}}) = p^1 = \langle \textbf{\textit{Bringing}}, \{ \langle \langle take \rangle, \textsc{Lu}, take \rangle, \ldots \} \rangle$$

for interpretation 5.5.

As already explained, the labeling process depends on linguistic information, as well as on the information derived from the $\mathcal{PM}$ (i.e., actions the robot is able to execute) and perceptual features extracted from the $\mathcal{P}^{\mathcal{PK}}$. In this Markovian framework, states reflect frame labels, and the decoding proceeds by detecting lexical units w_k to which the proper frame f^i is assigned. This association is represented as a pair $\langle w_k, f^i \rangle$, e.g., *take-**Taking***, *take-**Bringing***. A special null label "_" is used to express the status of all other words, e.g., *the*-_ or *mug*-_.

In the FD phase, each word is represented as a feature vector systematically defined to be a composition between linguistic, robot-dependent and environmental observations, as detailed below.

Linguistic features. Linguistic features here include lexical features (such as the surface or lemma of the current word and its left and right lexical contexts) and syntactic features (e.g., the Part-Of-Speech (POS)-tag of the current word or the contextual POS-tag n-grams).

Robot-dependent Features. Information about the robot coming from the $\mathcal{PM}$ are used to represent executable actions: these are mapped into frames through

their corresponding LUs. The $\mathcal{PM}$ thus defines a set of pairing between LUs and frames, according to which boolean features are used to suggest possibly activated frames for each word in a sentence. In particular, if w_k is a verb, and $F^k \subseteq F$ is the subset of frames that can be evoked by a word w_k (according to what stated in the $\mathcal{PM}$), then, for every frame $f^i \in F^k$, the corresponding i-th feature of the w_k is set to `true`.

Perceptual features. In addition, features derived from the operational context are used in the FD step as they are extracted from the $\mathcal{P}^{\mathcal{PK}}$. These "perception-based" features combine the information derived by the lexical grounding function with the syntactic dependency tree associated with s. In particular, let v_h be a verb. Let $n(v_h)$ be the set of nouns governed by the verb v_h, $n(v_h) = \{w_k \mid POS(v_h) == \text{VB} \wedge POS(w_k) == \text{NN} \wedge w_k \textit{ is rooted in } v_h \textit{ in the dependency (sub)tree}\}$. Let $t(v_h)$ be the set of tokens governed by the verb v_h, $t(v_h) = \{t_k \mid POS(v_h) == \text{VB} \wedge t_k \textit{ is rooted in } v_h \textit{ in the dependency (sub)tree}\}$. Then the following perceptual features are extracted and associated to each token of the sentence.

Grounded entities
The number $|n(v_h)|$ of nouns governed by v_h is added as a feature to the representation of all the tokens $t_k \in t(v_h)$. Even though this is not properly a perceptual evidence, its contribution must be considered when paired with another feature, whose aim is to explicits the number of entities that have been grounded by the tokens $w_k \in n(v_h)$. This feature is as well added to the representation of all the tokens $t_k \in t(v_h)$. Formally, its value is defined as the cardinality of the grounded sets union $\left| \bigcup_{\forall w_k \in arg_j^i \wedge w_k \in n(v_h)} \mathcal{G}_{arg_j^i} \right|$.

Spatial features
This is probably the key contributing feature among the perceptual ones. In fact, it tries to capture the spatial configuration of the involved entities populating the environment, by allowing an active control of the predicate prediction, whenever the distance between objects is the only discriminating factor. Operationally, $\forall w_k \in arg_j^i \wedge w_k \in n(v_h)$, their corresponding grounding sets $\mathcal{G}_{arg_j^i}$ are extracted. Then, from each $\mathcal{G}_{arg_j^i}$, the most promising candidate entities (i.e., the one maximizing g) are considered and the average Euclidean spatial distance between them is computed, by relying on the predicate `distance(e1, e2, d)`. The resulting feature is a discretized version of the averaged distance (i.e., `near`/`far`). Such a discrete value is obtained by comparing the Euclidean distance `d` against an empirically evaluated threshold ϵ.

Boundary Identification. For each identified predicate $p^i \in \mathcal{I}(s)$, the **Boundary Identification (BI)** step predicts all its arguments arg_j^i, by detecting the corresponding argument span a_j^i and semantic head h_j^i. This process starts filling the missing elements of each j-th argument $arg_j^i \in Arg^i$. More formally, for a given sentence s, the i^{th} identified predicate p^i, the BI process can be summarized as the

function $f_{BI}(s, p^i, \mathcal{P}^{\mathcal{PK}})$ updating the structure of $\mathcal{I}(s)$ as follows:

$$f_{BI}(s, p^i, \mathcal{P}^{\mathcal{PK}}) = p^1 = \langle \textbf{\textit{Taking}}, \{ \langle\langle take\rangle, \textsc{Lu}, take\rangle, \langle\langle the, mug, next, to, the, keyboard\rangle, \_, mug\rangle\}\rangle$$

for interpretation 5.4, or

$$f_{BI}(s, p^i, \mathcal{P}^{\mathcal{PK}}) = p^1 = \langle \textbf{\textit{Bringing}}, \{ \langle(take), \textsc{Lu}, take\rangle, \langle(the, mug), \_, mug\rangle, \langle(next, to, the, keyboard), \_, keyboard\rangle\}\rangle$$

for interpretation 5.5.

In the proposed Markovian framework, states now reflect argument boundaries between individual $arg^i_j \in Arg^i$. Following the `IOB2` notation, the Begin (`B`), Internal (`I`) or Outer (`O`) tags are assigned to each token. For example, the result of the BI over the sentence *"take the mug next to the keyboard"* would be

`O`-*take* `B`-*the* `I`-*mug* `I`-*next* `I`-*to* `I`-*the* `I`-*keyboard* (Interpr. 5.4)

or

`O`-*take* `B`-*the* `I`-*mug* `B`-*next* `I`-*to* `I`-*the* `I`-*keyboard* (Interpr. 5.5)

Linguistic features. In this step, the same morpho-syntactic features adopted for the FD are used together with the frame f^i detected in the previous step. For each token, its lemma, right and left contexts are considered as purely lexical features. Conversely, the syntactic features used are POS-tag of the current token and left and right contextual POS-tags n-grams.

Perceptual features. As for the FD step, the following dedicated features derived from the perceptual knowledge are introduced.

Grounded entities
For each noun $w_k \in arg^i_j$ such that $\mathcal{G}_{arg^i_j} \neq \emptyset$, a boolean feature is set to `true`. It is worth reminding that $\mathcal{G}_{arg^i_j}$ contains candidate entities referred by arg^i_j. Moreover, for each preposition $prep_k$, given their syntactic dependent $w^{dep}_k \in arg^i_j$, a boolean feature is set to `true` if and only if $\mathcal{G}_{arg^i_j} \neq \emptyset$. Again, for each preposition $prep_k$, the number of nouns $w_k \in arg^i_j$ on the left and on the right of $prep_k$, whose $\mathcal{G}_{arg^i_j} \neq \emptyset$, are also used as features in its corresponding feature vector.

Spatial features
For each preposition $prep_k$, we also retrieve its syntactic governor in the tree $w^{gov}_f \in arg^i_j$ and measure the average Euclidean distance in $\mathcal{P}^{\mathcal{PK}}$ between entities in $\mathcal{G}^{dep} \cup \mathcal{G}^{gov}$. A well as for the FD feature, if this score is under a given threshold ϵ, the spatial feature is set to `near`, replacing the default value of `far`.

Argument Classification. In the **Argument Classification (AC)** step, for each the frame $p^i = \langle f^i, Arg^i \rangle \in \mathcal{I}(s)$, all the $arg^i_j \in Arg^i$ are labeled according to their semantic role $r^i_j \in arg^i_j$, e.g., THEME to the argument *the mug next to the keyboard*, or THEME and GOAL to arguments *the mug* and *next to the keyboard*, respectively. In fact, in this step states correspond to role labels. Though some of the ideas have been already published in [17], the main novelty of this thesis is that classification here exploits both linguistic features and semantic information about the application domain extracted from the $\mathcal{P}^{\mathcal{DK}}$. This is possible thanks to the proposed framework, which allows to inject new features as they are identified as possibly contributing. Consequently, AC predictions will reflect also information extracted from the $\mathcal{P}^{\mathcal{DK}}$.

Given a predicate $p^i = \langle f^i, Arg^i \rangle$ and the class hierarchy $\mathcal{P}$, the AC function can thus be written as $f_{AC}(s, p^i, \mathcal{P}, DS)$ and produces the following complete structure

$$\begin{aligned} f_{AC}(s, p^i, \mathcal{P}, DS) \quad = p^1 = \quad & \langle \textbf{\textit{Taking}}, \{ \\ & \langle \langle take \rangle, \text{LU}, take \rangle, \\ & \langle \langle the, mug, next, to, the, keyboard \rangle, \text{THEME}, mug \rangle \} \rangle \end{aligned}$$

for interpretation 5.4, or

$$\begin{aligned} f_{AC}(s, p^i, \mathcal{P}, DS) \quad = p^1 = \quad & \langle \textbf{\textit{Bringing}}, \{ \\ & \langle (take), \text{LU}, take \rangle, \\ & \langle (the, mug), \text{THEME}, mug \rangle, \\ & \langle (next, to, the, keyboard), \text{GOAL}, keyboard \rangle \} \rangle \end{aligned}$$

for interpretation 5.5.

Linguistic features. Again, the same morpho-syntactic features adopted in both FD and BI are obtained from s, together with the frame p^i and the `IOB2` tags coming from the previous stages. For each token, its lemma, right and left contexts are considered as purely lexical features. The POS-tag of the current token and left and right contextual POS-tag n-grams are used as the syntactic features.

In addition, a model of DS is applied to generalize the argument semantic head h^i_j of each argument arg^i_j: the distributional (vector) representation for h^i_j is thus introduced to extend the feature vector corresponding to each $w_k \in a^i_j$, where a^i_j is a member of the triple $\langle a^i_j, r^i_j, h^i_j \rangle = arg^i_j \in Arg^i$, representing the argument span.

Domain-dependent features. Semantic features have been extracted from $\mathcal{P}^{\mathcal{DK}}$ to link the interpretation $\mathcal{I}(s)$ to the Domain Knowledge. However, grounded entities must be provided in order to extract such attributes from $\mathcal{P}^{\mathcal{DK}}$. Consequently, there is an implicit dependence of the AC on the $\mathcal{P}^{\mathcal{PK}}$. In particular, the following features have been designed to further generalize the model proposed in [17].

Entity-type property

The *Entity-type property* is a straightforward information that helps in generalizing the semantic head of an argument through the class the corresponding grounded

Feature	FD	BI	AC
Linguistic features	✓	✓	✓
Platform Model ($\mathcal{PM}$)	✓	✗	✗
Domain Knowledge ($\mathcal{P}^{\mathcal{DK}}$)	✗	✗	✓
Perception Knowledge ($\mathcal{P}^{\mathcal{PK}}$)	✓	✓	✗
Distributional Semantics (DS)	✗	✗	✓

Table 5.1. Feature modeling of the three steps (i.e., FD, BI and AC)

entity belongs to. Again, for each p^i and for each $arg^i_j \in Arg^i$, the semantic head h^i_j is grounded into a set of possible candidate entities through $\mathcal{G}_{arg^i_j}$. The most promising candidate `e`, i.e., $\max_e g(h^i_j, w_e)$, is extracted and its class `C`, obtained through the predicate `is-a(e,C)`, is applied to the semantic head feature vector.

Contain-ability property

The *Contain-ability property* is a domain-dependent semantic attribute, meaning that all the elements of `C` can contain something. To this end, for each p^i and for each $arg^i_j \in Arg^i$, the semantic head h^i_j is grounded into a set of possible candidate entities through $\mathcal{G}_{arg^i_j}$. The most promising candidate `e`, i.e., $\max_e g(h^i_j, w_e)$, is then extracted and a boolean feature is applied to the semantic head feature vector, reflecting the value of `is-contain-able(C,t)`, where `C` is the class the entity `e` belongs to.

A reader-friendly sum up is provided in Table 5.1 where, for each step of the processing cascade, features and resources used are shown. In particular, while BI uses only *Linguistic features* and $\mathcal{P}^{\mathcal{PK}}$, in FD even the $\mathcal{PM}$ is exploited. Conversely, as for the nature of the task, the AC step mostly relies on $\mathcal{P}^{\mathcal{DK}}$ and DS, in order to provide effective generalization capability while choosing the correct semantic role.

5.3. Experimental Evaluation and Results

The scalability of the proposed framework towards the systematic introduction of perceptual information has been evaluated in the semantic interpretation of utterances in a house Service Robotics scenario. The evaluation is carried out using the Human-Robot Interaction Corpus (HuRIC), presented in 5.4, that contains commands in two languages, i.e., English and Italian ([168]).

The DS vectors used in the grounding function Γ have been acquired through a Skip-gram model [115], through the `word2vec` tool (see Appendix A.1.3). By applying the settings *min-count=50*, *window=5*, *iter=10* and *negative=10* onto the UkWaC corpus [51], 250 dimensional word vectors have been derived for more than $110,000$ words. The SVM^{hmm} algorithm has been implemented within the KeLP framework [52].

Measures have been carried out on four tasks, according to a 10-fold evaluation schema. The first three correspond to evaluating the individual interpretation steps, namely the FD, BI and AC, (Sections 5.3.1, 5.3.2 and 5.3.3). In these tests, gold annotations are assumed as input information for the task, even if they depend on a previous processing step. The last test (Section 5.3.4) concerns the analysis of the

		FD			
		P	**R**	**F1**	**RER**
EN	*pLing*	$94.52\% \pm 0.04$	$94.32\% \pm 0.08$	$94.41\% \pm 0.05$	-
	Ground	$95.59\% \pm 0.02$	$96.31\% \pm 0.05$	$95.94\% \pm 0.03$	27.42%
IT	*pLing*	$94.84\% \pm 0.22$	$95.58\% \pm 0.19$	$95.19\% \pm 0.19$	-
	Ground	$95.14\% \pm 0.17$	$95.54\% \pm 0.15$	$95.32\% \pm 0.14$	2.52%

Table 5.2. FD results: evaluating the whole span

end-to-end interpretation chain. It thus corresponds to the ability of interpreting a fully grounded and executable command and reflects the behavior of the system in a real scenario.

While $\mathcal{P}^{\mathcal{PK}}$ is involved in both the FD and BI tasks, AC relies just on the $\mathcal{P}^{\mathcal{DK}}$ and the DS. Hence, in order to emphasize the contribution of such information, two settings have been tested.

The first relies just on linguistic features and information from the Semantic Map is neglected. We call this setting *Pure Linguistic* (*pLing*), as the interpretation is driven just by lexical/syntactic observation of the sentence. It refers to a configuration in which only the features corresponding to the first two rows of Table 5.1 are considered.

The second is a *Grounded* (*Ground*) setting. It is built upon the features designed around the Semantic Map, that has been encoded into a set of predicates $\mathcal{P}$, and DS, represented by Word Embeddings. In order to enable for the extraction of meaningful properties from $\mathcal{P}$, grounding is based on the set $\mathcal{G}$ of entities populating the environment and is built using the grounding function $\Gamma(arg_j^i, \mathcal{P}^{\mathcal{PK}})$. $\mathcal{P}^{\mathcal{PK}}$ features are injected into the FD and BI steps, while $\mathcal{P}^{\mathcal{DK}}$ features together with Word Embeddings are used into the AC process. Hence, this setting applies all the features defined in Table 5.1.

Results obtained in every run are reported in terms of Precision (P), Recall (R) and F-Measure (F1) as a micro-statistics across the 10 folds. The contribution of Semantic Map information is emphasized in terms of Relative Error Reduction (RER) over F1 with respect to the *pLing* setting, relying just on linguistic information.

5.3.1. Frame Detection

This experiment allows evaluating the performance of the system in recognizing the actions evoked by the command. This step represents the entry point of the interpretation cascade: minimizing the error at this stage is essential to avoid error propagation throughout the whole pipeline.

Table 5.2 reports the results obtained for the two settings *pLing* and *Ground*, over the two datasets (i.e., English and Italian). In this case, we count a prediction as correct only whenever all the tokens belonging to *lexical unit* (LU) have been correctly classified.

First, it is worth emphasizing that the F1 is always higher than 94%. This means that the system will be (almost) always able to detect the correct action expressed by the command. In fact, linguistic features seem to already model the problem with a good coverage of the phenomena.

		BI			
		P	**R**	**F1**	**RER**
EN	*pLing*	$89.62\% \pm 0.11$	$91.61\% \pm 0.03$	$90.59\% \pm 0.05$	-
	Ground	$90.04\% \pm 0.16$	$91.33\% \pm 0.10$	$90.67\% \pm 0.12$	0.86%
IT	*pLing*	$82.89\% \pm 0.84$	$85.51\% \pm 0.58$	$84.14\% \pm 0.68$	-
	Ground	$83.41\% \pm 0.84$	$86.30\% \pm 0.56$	$84.77\% \pm 0.66$	4.02%

Table 5.3. BI results: evaluating the whole span

However, when perceptual features (extracted from the $\mathcal{P}^{\mathcal{PK}}$) are injected, the F1 increases up to 95.94%, with a RER of 27.42%. The contribution of such evidence is mainly due to one of the most frequent errors, concerning the ambiguity of the *"take"* verb. In fact, as explained in Section 5.2, due to the PP attachment ambiguity, the interpretation of such verb may be different (i.e., either ***Bringing*** or ***Taking***, depending on the spatial configuration of the environment. As the *pLing* setting does not rely on any kind of perceptual knowledge, the system is not able to correctly discriminate among them. Hence, the resulting interpretation is more likely to be wrong, as it does not reflect the semantics carried by the environment.

On the other hand, the Italian dataset does not seem to benefit from these features. In fact, the RER in such a configuration is 2.52% (i.e., from 95.19% to 95.32%). This is probably due to the absence of the above linguistic phenomena into the Italian language.

5.3.2. Boundary Identification

In this section, the ability of the BI classifier in identifying the argument spans of the commands' predicates is evaluated. According to the results reported in Table 5.3, this task seems to be the most challenging one.

In fact, the F1 settles just under the 91% on the English dataset, with the *pLing* and *Ground* settings scoring 90.59% and 90.67% respectively. Moreover, in this case $\mathcal{P}^{\mathcal{PK}}$ does not seem to substantially contribute to the correct classification of the argument boundaries.

On the other hand, in the Italian setting the F1 does not exceed 85% (84.14% and 84.77% for the *pLing* and *Ground* settings). However, the perceptual information contributes to a slightly larger gain with respect to the one obtained on English. This is probably due to the presence of commands where the spatial configuration of the environment is essential to correctly chunk the argument spans. For example, for a command like *"porta il libro sul tavolo in cucina"* (*"bring the book on the table in the kitchen"*), the fragment *il libro sul tavolo* (*the book on the table*) may correspond to one single argument in which *sul tavolo* (*on the table*) is a spatial modifier of *il libro* (*the book*). In this case, *in cucina* (*in the kitchen*) composes another semantic argument. This interpretation is spatially correct whenever, within the corresponding Semantic Map, *the book* is *on the table* and the latter is outside the *kitchen*. Conversely, if *the book* is not *on the table* which is, in turn, into *the kitchen*, then *sul tavolo in cucina* (*on the table in the kitchen*) will constitute an entire argument span.

		AC			
		P	**R**	**F1**	**RER**
EN	*pLing*	94.46% ± 0.05	94.46% ± 0.05	94.46% ± 0.05	-
	Ground	95.49% ± 0.05	95.49% ± 0.05	95.49% ± 0.05	18.65%
IT	*pLing*	91.52% ± 0.23	91.52% ± 0.23	91.52% ± 0.23	-
	Ground	92.21% ± 0.11	92.21% ± 0.11	92.21% ± 0.11	8.14%

Table 5.4. AC results: evaluating the whole span

5.3.3. Argument Classification

This experiment is the most interesting one, as here we inject the novel information with respect to [17], extracted from $\mathcal{P}^{\mathcal{DK}}$, regarding the *Contain-ability* property and the class of the grounded entity.

As reported in Table 5.4, the system is able to recognize the involved entities with high accuracy, with an F1 higher than 91.50% in both the English and Italian datasets. This result is surprising when analyzing the complexity of the task. In fact, the classifier is able to cope with a high level of uncertainty, as the amount of possible semantic roles is sizable, i.e., 34 for the English dataset, 27 for the Italian one.

Beside obtaining outstanding accuracy in all the configurations, a twofold contribution is achieved when distributional information about words and domain specific evidence is adopted. On the one hand, DS injects beneficial lexical generalization into training data: frame elements of arguments whose semantic heads are close in the vector space are seemingly tagged. For example, if *the book* in the training sentence "*take the book*" is the THEME of a ***Taking*** frame, similar arguments for the same frame will receive the same role label as *volume* in "*grab the volume*". Moreover, further lexical generalization is provided by including the class name of the grounded entity in the feature space, so that lexical references like *tv*, *tv set*, *television set* and *television* all refer to the same class *Television*.

On the other hand, information related to domain-dependent attributes of a given class might be helpful to solve specific errors of the AC process. For example, when including the *Contain-ability* property as a feature, we are implicitly suggesting to the learning function that an object can contain something. Consequently, this information allows to better discriminate whether an object must be labeled as *"Containing_object"* rather than *"Container_portal"*.

5.3.4. End-to-End Processing Cascade

This section concludes the experimental evaluation by reporting the results obtained through the end-to-end processing cascade. In this case, each step is fed with the labels coming from the previous one: it thus represents a real scenario configuration, when the system is operating on a robot.

In this configuration, only the results of the AC step are reported (Table 5.4), as its output represents the end of the pipeline. Moreover, in this setting, the error propagation is implicitly estimated, as each step is fed the information output from the previous one. These results give thus an idea of the performance of the whole

		P	R	F1	RER
			AC		
EN	*pLing*	86.12% ± 0.16	81.41% ± 0.29	83.67% ± 0.22	-
	Ground	89.25% ± 0.11	86.39% ± 0.22	87.77% ± 0.14	25.10%
IT	*pLing*	77.10% ± 0.81	76.08% ± 0.80	76.47% ± 0.72	-
	Ground	78.33% ± 0.85	77.23% ± 0.53	77.67% ± 0.60	5.09%

Table 5.5. Evaluating the end-to-end chain against the whole span

		P	R	F1	RER
			AC		
EN	*pLing*	91.04% ± 0.07	91.54% ± 0.07	91.28% ± 0.06	-
	Ground	92.90% ± 0.04	93.34% ± 0.04	93.11% ± 0.02	20.89%
IT	*pLing*	83.07% ± 0.41	87.30% ± 0.30	85.07% ± 0.31	-
	Ground	84.15% ± 0.33	88.83% ± 0.27	86.35% ± 0.24	8.58%

Table 5.6. Evaluating the end-to-end chain against the semantic head

system. Note that the DS and the domain-dependent features (*Ground* setting) boost the performance for both languages. More specifically, the *Ground* configuration consistently outperforms the *pLing* one for English, suggesting the benefits given by the promoted feature space. For Italian, this behavior is less evident, even tough results confirm the general trend.

In order to provide an even more realistic evaluation of the system, the performance of the system has been measured by considering only the prediction over the semantic heads. This evaluation wants to reproduce the usage of the framework, where just the semantic head is adopted to instantiate and execute a plan. For example, given the command *"take the mug next to the keyboard"*, together with one of its interpretations

$$[take]_{\textbf{\textit{Taking}}}\ [the\ mug\ next\ to\ the\ keyboard]_{\text{THEME}},$$

only two information are required in order for the robot to execute the requested action, namely the type of the action ***Taking*** and the object to be taken, *mug*, which is pointed by the semantic head of the THEME argument.

The results reported in Table 5.5 are extremely encouraging for the application of the proposed framework in realistic scenarios. In fact, over the English dataset, the F1 is always higher than 91% in the recognition of the correct label of the semantic head, along with semantic predicates and boundaries used to express intended actions. Moreover, the recognition of the full command benefits from Semantic Map features, with an F1 score increasing to 93.11%. In addition, the low variance suggests a good stability of the system against the random selection of the training/tuning/testing sets.

Though with lower results, such a trend is confirmed over the Italian dataset. In fact, the difference between the two datasets is owed by two reasons: first, the different linguistic phenomena and ambiguities present in the two languages do not allow to directly compare the two empirical evaluations; second, the small number of examples used to train/test the models biases the final results, provided that the

	English	Italian
Number of examples	656	241
Number of frames	18	14
Number of predicates	762	272
Number of roles	34	28
Predicates per sentence	1.16	1.13
Sentences per frame	36.44	17.21
Roles per sentence	2.02	1.90
Entities per sentence	6.59	6.97

Table 5.7. HuRIC: some statistics

Italian dataset is composed of only 241 commands. However, the system seems to be deployable on a real robot, with the best configuration obtaining an F1 of 86.36%.

5.4. HuRIC - Human-Robot Interaction Corpus

The computational paradigms proposed in this chapter are based on machine learning techniques that strictly depend on the availability of training data. Hence, in order to properly train and test the language understanding framework, a collection of datasets has been developed, that together form the HuRIC[3], formerly presented in [13]. Since its first release, the corpus has been extended including several dimensions [169]. This section provides a detailed presentation of the resource.

HuRIC is based on Frame Semantics ([53]) and captures cognitive information about situations and events expressed in sentences. The most interesting feature is that HuRIC is not system or robot dependent, both with respect to the surface of sentences and with respect to the adopted formalism. In fact, HuRIC contains information strictly related to NL semantics and it thus results decoupled from the specific system.

Each sentence in HuRIC is then annotated with: *lemmas*, *POS tags*, *dependency trees* and *FrameNet* annotations ([6]). Semantic frames and frame elements are used to represent the meaning of commands, as, in our view, they reflect the actions a robot can accomplish in a home environment. In this way, HuRIC can potentially be used to train all the modules of the processing chain presented in Section 5.2.3.

HuRIC provides commands in two different languages: English and Italian. While the English subset contains 656 sentences, 241 commands are available in Italian ([168]). The number of annotated sentences, number of frames and further statistics are reported in Table 5.7. Almost all Italian sentences are translations of the original commands in English and the corpus maintains the alignment between those sentences. These alignments might support further researches in different areas, such as in the context of Machine Translation. Detailed statistics about the number of sentences for each frame and frame elements are reported in Table 5.8 and 5.9 for the English and Italian subsets, respectively.

The corpus exploits different situations representing possible commands given to a robot in a house environment. Hence, examples collected in HuRIC range

[3] Available at `http://sag.art.uniroma2.it/huric`.

Frame	Ex	Frame	Ex	Frame	Ex
Motion	143	***Bringing***	153	***Cotheme***	39
Goal	129	Theme	153	Cotheme	39
Theme	23	Goal	95	Manner	9
Direction	9	Beneficiary	56	Goal	8
Path	9	Agent	39	Theme	4
Manner	4	Source	18	Speed	1
Area	2	Manner	1	Path	1
Distance	1	Area	1	Area	1
Source	1				
Locating	90	***Inspecting***	29	***Taking***	80
Phenomenon	89	Ground	28	Theme	80
Ground	34	Desired_state	9	Source	16
Cognizer	10	Inspector	5	Agent	8
Purpose	5	Unwanted_entity	2	Purpose	2
Manner	2				
Change_direction	11	***Arriving***	12	***Giving***	10
Direction	11	Goal	11	Recipient	10
Angle	3	Path	5	Theme	10
Theme	1	Manner	1	Donor	4
Speed	1	Theme	1	Reason	1
Placing	52	***Closure***	19	***Change_operational_state***	49
Theme	52	Containing_object	11	Device	49
Goal	51	Container_portal	8	Operational_state	43
Agent	7	Agent	7	Agent	17
Area	1	Degree	2		
Being_located	38	***Attaching***	11	***Releasing***	9
Theme	38	Goal	11	Theme	9
Location	34	Item	6	Goal	5
Place	1	Items	1		
Perception_active	6	***Being_in_category***	11	***Manipulation***	5
Phenomenon	6	Item	11	Entity	5
Manner	1	Category	11		

Table 5.8. Distribution of frames and frame elements in the English dataset

Frame	Ex	Frame	Ex	Frame	Ex
Motion	51	***Locating***	27	***Inspecting***	4
Goal	28	Phenomenon	27	Ground	2
Direction	20	Ground	6	Unwanted_entity	2
Distance	13	Manner	2	Desired_state	2
Speed	8	Purpose	1	Instrument	1
Theme	3				
Path	2				
Manner	1				
Source	1				
Bringing	59	***Cotheme***	13	***Placing***	18
Theme	60	Cotheme	13	Theme	18
Beneficiary	31	Manner	6	Goal	17
Goal	26	Goal	5	Area	1
Source	8				
Closure	10	***Giving***	7	***Change_direction***	21
Container_portal	6	Theme	7	Direction	21
Containing_object	5	Recipient	6	Angle	9
Degree	1	Donor	1	Speed	9
Taking	22	***Being_located***	14	***Being_in_category***	4
Theme	22	Location	14	Item	4
Source	8	Theme	12	Category	4
Releasing	8	***Change_operational_state***	14		
Theme	8	Device	14		
Place	3				

Table 5.9. Distribution of frames and frame elements in the Italian dataset

from simple and direct commands like "*go to the kitchen*", to more complex ones like "*robot can you please go to the living room turn right and turn off the tv*". HuRIC is composed of different subsets (i.e., 7 for the English and 4 for the Italian subsets), characterized by different order of complexity, designed to differently stress a labeling architecture. Each dataset includes a set of audio files representing robotic commands, paired with the correct transcription. The environmental conditions in which the commands have been uttered, are homogeneous within each subset, ranging from very noisy, to completely noise-free.

The current release of HuRIC is made available through an XML-based format, whose extension is `.hrc` (Listing 5.1).

```
<?xml version="1.0" encoding="UTF-8"?>
<huricExample id="2650">
  <commands>
    <command>
      <sentence>take the mug next to the keyboard</sentence>
      <tokens>
        <token id="1" lemma="take" pos="VB" surface="take" />
        <token id="2" lemma="the" pos="DT" surface="the" />
        <token id="3" lemma="mug" pos="NN" surface="mug" />
        <token id="4" lemma="next" pos="JJ" surface="next" />
        <token id="5" lemma="to" pos="TO" surface="to" />
        <token id="6" lemma="the" pos="DT" surface="the" />
        <token id="7" lemma="keyboard" pos="NN" surface="keyboard" />
      </tokens>
      <dependencies>
        <dep from="0" to="1" type="root" />
        <dep from="1" to="3" type="dobj" />
        <dep from="3" to="2" type="det" />
        <dep from="1" to="4" type="advmod" />
        <dep from="4" to="7" type="nmod" />
        <dep from="7" to="5" type="case" />
        <dep from="7" to="6" type="det" />
      </dependencies>
      <semantics>
        <frames>
          <frame name="Bringing">
            <lexicalUnit>
              <token id="1" />
            </lexicalUnit>
            <frameElements>
              <frameElement type="Theme">
                <token id="2" />
                <token id="3" />
              </frameElement>
              <frameElement type="Goal">
                <token id="4" />
                <token id="5" />
                <token id="6" />
                <token id="7" />
              </frameElement>
            </frameElements>
          </frame>
        </frames>
      </semantics>
      <audioFiles>
```

```
46        <file name="newCorpus97_daniele.flac" />
47      </audioFiles>
48    </command>
49  </commands>
50  <semanticMap>
51    <entities>
52      <entity atom="p1" type="Cup">
53        <attributes>
54          <attribute name="contain_ability">
55            <value>true</value>
56          </attribute>
57          <attribute name="lexical_references">
58            <value>cup</value>
59            <value>mug</value>
60            <value>coffee cup</value>
61            <value>bowl</value>
62          </attribute>
63        </attributes>
64        <coordinate angle="0.0" x="2.0" y="5.0" z="0.0" />
65      </entity>
66      ...
67      <entity atom="k1" type="Keyboard">
68        <attributes>
69          <attribute name="contain_ability">
70            <value>false</value>
71          </attribute>
72          <attribute name="lexical_references">
73            <value>keyboard</value>
74            <value>console</value>
75          </attribute>
76        </attributes>
77        <coordinate angle="0.0" x="4.0" y="1.0" z="0.0" />
78      </entity>
79    </entities>
80  </semanticMap>
81  <lexicalGroundings>
82    <lexicalGrounding atom="p1" tokenId="3" />
83    <lexicalGrounding atom="k1" tokenId="7" />
84  </lexicalGroundings>
85</huricExample>
```

Listing 5.1. Excerpt of an hrc file

Hence, for each command, the following information are stored:

1. the whole sentence (i.e., line 5);
2. the list of tokens composing it, along with the corresponding lemma and POS tag (i.e., lines 6-14);
3. the dependency relations among tokens (i.e., lines 15-23);
4. the semantics, expressed in terms of Frames and Frame elements (i.e., lines 24-40);
5. the audio files associated to the command (i.e., lines 41-43);

6. the configuration of the environment, in terms of entities populating the Semantic Map, along with their semantic attributes (i.e., lines `46-74`);

7. the gold groundings, providing mapping between linguistic symbols (namely, words of the sentence) and entities of the semantic map (i.e., lines `75-78`).

The main extension of HuRIC with respect to the version presented in [13] is represented by pairing each utterance with a possible reference environment ([169]). Each command is thus provided with an automatically generated Semantic Map, reflecting the disposition of entities matching the interpretation, so that perceptual features can be consistently derived for each command; hence, the latter can be interpreted with respect to the environment itself. The map generation process has been designed to reflect real application conditions. First, the $\mathcal{P}^{\mathcal{DK}}$ described in Section 2.4 has been used to describe our world, in terms of classes (or categories) that refer to entities of a generic home environment. Then, for each sentence s, the corresponding $\mathcal{P}^{\mathcal{PK}}$ is populated with the set of entities referred by the sentence, plus a control set of 20 randomly-generated additional objects, all taken from the $\mathcal{P}^{\mathcal{DK}}$ specification. The naming set `N` of each class has been defined by simulating the lexical references introduced through a process of HAM. To this end, for every class name in the $\mathcal{P}^{\mathcal{DK}}$, a range of possible polysemic variations has been defined, by automatically exploiting lexical resources, such as WordNet [116], or by corpus-analysis. The final set has been then validated by human annotators. As an example, the class `Cup` is referred through the following variations: *cup*, *mug*, *coffee cup* and *bowl*. The lexical variation allows augmenting the data set, as each training sentence can be paired with more than one $\mathcal{P}^{\mathcal{PK}}$.

5.5. The LU4R framework: adaptive spoken Language Understanding For Robots

The computational paradigms presented in this chapter have been implemented and made available to the community through a tool called LU4R ([18, 169]).[4] In fact, LU4R embodies the capabilities in terms of linguistic generalization characterizing the presented data-driven approach. Such system has been already tested and deployed in real robots, both in service robotics [172] and industrial applications [45].

The architecture of the LU4R framework considers two main actors, as shown in Figure 5.4: the *Robotic Platform* and the *LU4R chain* (or LU4R). The communication between the robot and the chain is realized through a Client/Server architecture, where the Robotic Platform is the Client, whereas LU4R is the Server. The Client-Server communication scheme between LU4R and the Robot allows for the independence from the Robotic Platform, in order to maximize the re-usability and integration in heterogeneous robotic settings.

As explained in Section 5.2, the SLU process implemented exhibits semantic capabilities (e.g., disambiguation, predicate detection or grounding into robotic actions and environments) that are designed to be general enough to be representative of a large set of application scenarios. On the one hand, it is obvious that an

[4] Available at `http://sag.art.uniroma2.it/lu4r.html`

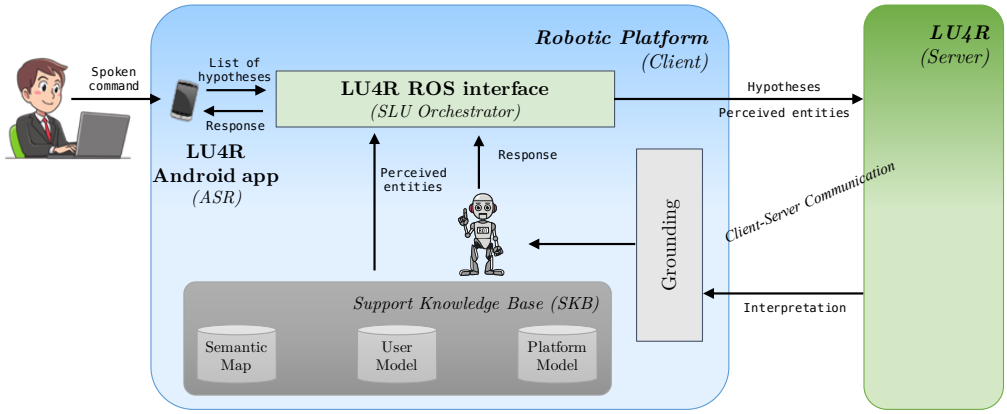


Figure 5.4. The LU4R framework architecture

interpretation process must be achieved even when no contextual information about the domain/environment is available, i.e., a scenario involving a *blind*, but speaking robot, or when the actions a robot can perform are not made explicit. This is the case when the command *"take the mug next to the keyboard"* is not paired with any additional information and the ambiguity with respect to the evoked frame, i.e., ***Taking*** vs. ***Bringing***, cannot be resolved. On the other hand, LU4R makes available methods to specialize its semantic interpretation process to individual situations where more information is available about goals, the environment, and the robot capabilities. These methods are expected to support the optimization of the core SLU process against a specific interactive robotics setting, in a cost-effective manner. In fact, whenever more information about the perceived environment (e.g., a semantic map) or about robot capabilities is provided, the interpretation of a command can be improved by exploiting a more focused scope. That is, whenever the sentence *"take the mug next to the keyboard"* is provided along with information about the presence and position of `mug` and `keyboard`, the system is able to detect the intended action, i.e., again either ***Taking*** or ***Bringing***.

In order to better describe the different operating modalities of LU4R, some assumptions toward the Robotic Platform must be made explicit; this allows to precisely establish functionalities and resources that the robot needs to provide to unlock the more complex processes. This information is thus used to express the experience that the robot is able to share with the user (i.e., the perceived environment where the linguistic communication occurs, and some lexical and semantic properties about entities populating the environment) and some level of awareness about its own capabilities (e.g., the primitive actions that the robot is able to perform, given its hardware components). In the following, each component of the architecture in Figure 5.4 is discussed and analyzed.

5.5.1. The Robotic Platform

The LU4R framework contemplates a generic Robotic Platform, whose task, domain and physical setting are not necessarily specified. In order to make the SLU process independent from the above specific aspects, the platform is assumed to require, at

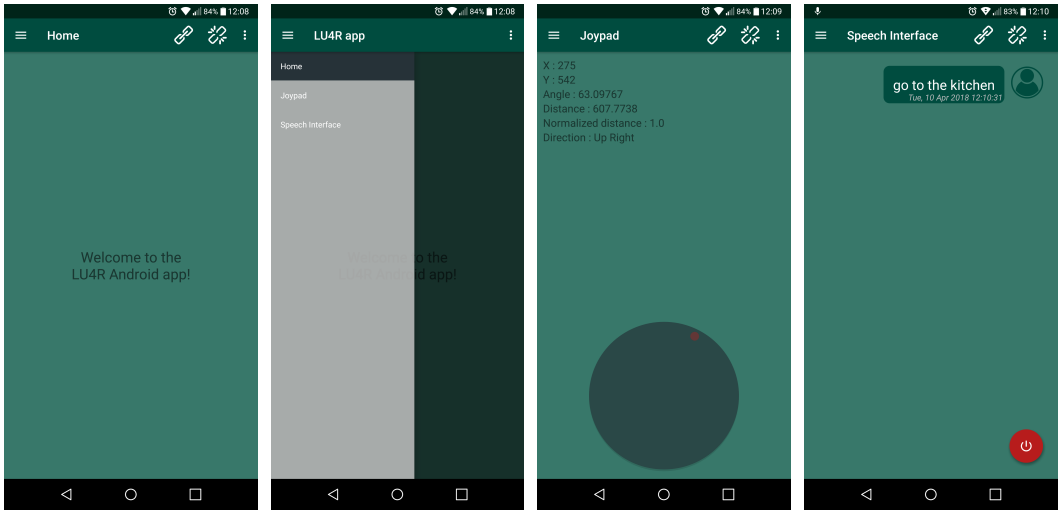


Figure 5.5. The LU4R Android app

least, the following modules:

- an Automatic Speech Recognition (ASR) system;
- an SLU Orchestrator;
- a Grounding and Command Execution Engine;
- a Physical Robot.

In developing the LU4R framework, both the ASR system and an SLU Orchestrator have been implemented. The ASR is realized through the ad-hoc *LU4R Android app*, whereas the *SLU Orchestrator* is implemented as a bunch of Robot Operating System (ROS) nodes, here collected in a single component called *LU4R ROS interface*. Additionally, the optional component *Support Knowledge Base (SKB)* is expected to maintain and provide the contextual information discussed above. Such resource aims at collecting some of the components presented in Section 5.2.1 (Figure 5.2).

While the discussion of the Robotic Platform is out of the scope of this thesis, all the other components are hereafter shortly summarized.

LU4R Android app. An ASR engine allows to transcribe a spoken utterance into one or more transcriptions. In the LU4R framework, the ASR is performed through an *ad-hoc* Android application, the LU4R Android app.[5] Figure 5.5 shows some of the capabilities of the LU4R Android app.

It relies on the official *Google ASR APIs*, that offer valuable performances for an off-the-shelf solution. The free-form ASR can be executed either in a continuous recognition setting or in a push-to-talk configuration. The main requirement of this solution is that the device hosting the software must feature an Internet connection, in order to provide transcriptions for the spoken utterance. The App can be deployed

[5] Available at `https://gitlab.com/lu4r_utilities/lu4r_android_app`.

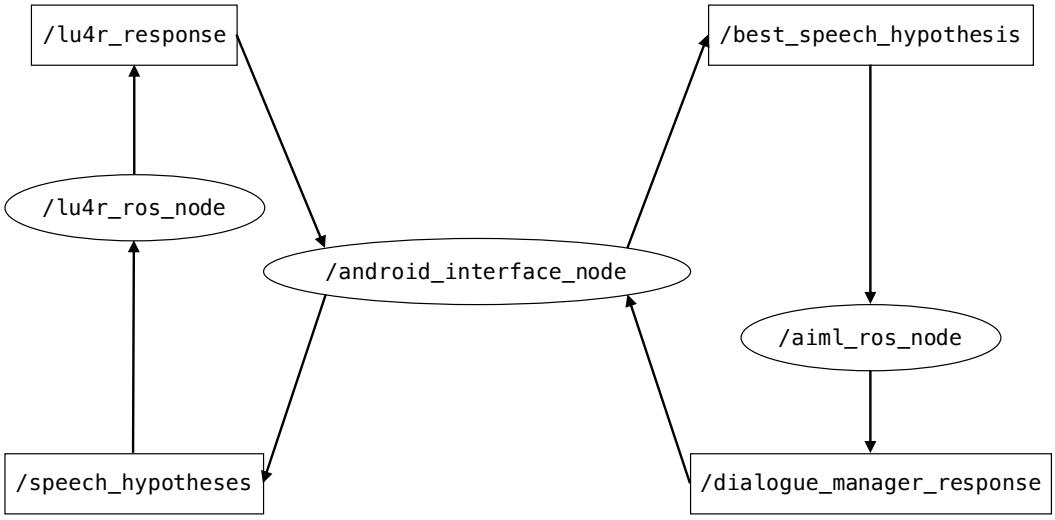


Figure 5.6. ROS computation graph of the LU4R ROS interface

on both Android smart-phones and tablets. In the latter case, even though the communication protocol remains the same, the tablet will be part of the robotic platform. The tablet can be provided with a directional condenser microphone and speakers.

The communication with the entire system is realized through TCP Sockets. In this setting, the LU4R Android app implements a TCP Client, feeding LU4R with lists of hypotheses through a middle-layer. To this end, the LU4R ROS interface has been integrated in the loop, acting as the TCP Server.

Once a new sentence is uttered by the user, this component outputs a list of hypothesized transcriptions, that are forwarded to the LU4R ROS interface.

In addition, the LU4R Android app features a virtual joypad, for tele-operating the robot. In this way, this tool provides a complete system for controlling a robotic platform.

LU4R ROS interface. As already stated, the LU4R ROS interface is a collection of ROS nodes/packages that enable a full integration of LU4R into the ROS environment. The communication between the nodes leverages the ROS *publisher/subscriber protocol* over *topics*, as shown in Figure 5.6. In this way, the LU4R ROS interface can be deployed into any ROS-based robotic platform. The LU4R ROS interface is thus composed of the following modules:

- `android_interface` is the main orchestrator of the LU4R ROS interface;
- `lu4r_ros` provides an interface to LU4R;
- `aiml_ros` implements a simple Dialogue Manager (DM), coded as an Artificial Intelligence Markup Language (AIML) Knowledge Base (KB);
- `framenet_ros_msgs` provides a mapping between FrameNet frames and ROS messages.

Thanks to the modular architecture, each component is designed to be easily replaced; the overall system described in the following.

android_interface. The `android_interface`[6] is the entry point of the system, acting as the actual orchestrator of the framework. It implements a TCP Server for the LU4R Android app and is coded as a python ROS node waiting for Client requests. Once a new request is received (i.e., a list of transcriptions for a given spoken sentence from the LU4R Android app), this module is in charge of extracting the perceived entities from a structured representation of the environment (i.e., the Semantic Map, a sub-component of the SKB). Then, whenever a running instance of LU4R is detected, the full list of hypotheses is published on a dedicated topic (i.e., `/speech_hypotheses`). Otherwise, the best hypothesis is published on the `/best_speech_hypothesis`.

This node is subscribed to two topics: (i) `/lu4r_response`, containing the feedback provided by LU4R in terms of FrameNet frames, and (ii) `/dialogue_manager_response` that, instead, hosts the reply of a Dialogue Manager. Depending on the response, the `android_interface` is then allowed to send a message back to the LU4R Android app.

The communication protocol requires the serialization of both speech hypotheses and entities of the Semantic Map into two different JSON objects (see Section 5.5.2 for more details). However, in order to obtain the desired interpretation, only the list of transcription is mandatory. In fact, even though the environmental information is essential for the perception-driven interpretation, whenever it is not provided, the chain operates in a blind setting.

In addition, such node interfaces with a ROS-compliant navigation system; hence, through the virtual joypad coded into the Android App (Figure 5.5), it is possible to tele-operate the robot.

lu4r_ros. The `lu4r_ros`[7] is an interface to LU4R, providing the latter with transcribed sentences and retrieving interpretations, through HTTP communications. When launching the node, it requires the IP address and port of a running LU4R instance; then, the node waits until LU4R is ready to serve.

This ROS node is subscribed to `/speech_hypotheses`, which contains the hypotheses list, encoded as a JSON string (see Section 5.5.2). This string is then sent to LU4R, whose reply is published onto the `/lu4r_response`, as interpretation encoded in Command Frame Representation (CFR) format [146].

aiml_ros. The `aiml_ros`[8] implements a DM through AIML KBs. This node depends on `PyAIML`[9], an interpreter designed to correctly handle AIML files.

The node is subscribed to the `/best_speech_hypothesis` topic, containing the best transcription provided by the LU4R Android app. Whenever a new transcription is published onto the topic, a callback function processes the user's

[6] Available at `https://gitlab.com/andreavanzo/framenet_ros_msgs`.
[7] Available at `https://gitlab.com/andreavanzo/lu4r_ros`
[8] Available at `https://gitlab.com/andreavanzo/aiml_ros`
[9] Available at `https://pypi.python.org/pypi/PyAIML`

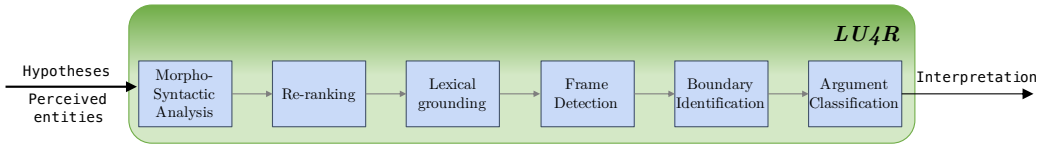


Figure 5.7. The LU4R interpretation cascade

utterance, gathering the next dialogue turn from the AIML KB. The result is finally published onto the `/dialogue_manager_response`.

`framenet_ros_msgs`. The `framenet_ros_msgs`[10] provides a mapping between FrameNet frames and ROS messages for a better integration of the linguistic theory into the ROS environment. Each frame is encoded into a ROS message, with semantic arguments corresponding to ROS message fields. For example, the THEME semantic argument of the ***Bringing*** frame is mapped into the field `string theme`.

Grounding and Command Execution. Even though the grounding process is placed at the end of the loop, it is discussed here, as it is a component of the Robotic Platform. In fact, this process has been completely decoupled from the SLU process, as it may involve perception capabilities and information unavailable to LU4R or, in general, out of the linguistic dimension. Nevertheless, this situation can be partially compensated by defining mechanisms to exchange some of the grounding information with the linguistic reasoning component. The grounding carried out by the robot is triggered by a logical form expressing one or more actions through logic predicates, that potentially correspond to specific frames. The output of LU4R embodies the produced logic form: this latter exposes the recognized actions that are then linked to specific robotic operations (i.e., primitive actions or plans). Correspondingly, the predicate arguments (e.g., objects and location involved in the targeted action) are detected and linked to the objects/entities of the current environment. A fully grounded command is obtained through the complete instantiation of the robot action (or plan) and its final execution.

5.5.2. The LU4R component

The LU4R component implements the language understanding cascade (Figure 5.7) described in Section 5.2.3, whose models have been trained over the HuRIC corpus (see Section 5.4). It realizes the interpretation service as a black-box component, so that the complexity of each inner sub-task is hidden to the user. The service is realized through a server that keeps listening to natural language sentences and outputs the corresponding interpretation. It is entirely coded in Java and released as a single Jar file. The LU4R module is composed by six modules, whose final output is the interpretation of a utterance. First, **Morpho-syntactic and syntactic analysis** is performed over the available utterance transcriptions by applying morphological analysis, POS tagging and syntactic analysis. In particular, dependency trees are extracted from the sentence as well as POS tags. Morpho-syntactic and syntactic

[10] Available at `https://gitlab.com/andreavanzo/framenet_ros_msgs`.

analysis is realized through the Stanford CoreNLP suite [110] when English is the targeted language, and the *Chaos* parser [9] for Italian commands.

Then, if more than one transcription hypothesis is available, the **Re-ranking** module is activated to compute a new ranking of the hypotheses list, in order to get the best transcription out of the initial ranking. This module is realized through two orthogonal approaches: a *learn-to-rank* approach, where a SVM exploiting a combination of linguistic Kernels is applied [10], or a domain-dependent approach, where grammar designed for common HRI tasks is leveraged to improve the robustness of the ASR through a *scaling-down* strategy (see Chapter 4).

The **Linguistic Grounding** module implements the Linguistic Grounding function described in Section 5.2.2. It aims at providing candidate grounded entities, extracted from the Perception Knowledge, for the lexical symbols of the sentence. This information is then injected into the subsequent steps as features, for linking the interpretation to the context of the sentence.

Finally, the best transcription along with grounded entities are the inputs of the interpretation cascade presented in Section 5.2.3 and composed of **Frame Detection**, **Argument Identification** and **Argument Classification** steps. The SVM^{hmm} algorithm for the above three steps of the semantic analysis is implemented through the KeLP [52].

LU4R is a service that can be invoked through HTTP requests. Its implementation is realized through a server that keeps listening to NL sentences and outputs an interpretation for them. The communication between the client of the service (the Robotic Platform) and LU4R is described hereafter. The LU4R Chain requires an **initialization phase**, where the process is run and initialized, followed by a **service phase**, where LU4R is ready to receive requests.

Initialization phase. The initialization phase consists in creating an instance of the server, among the ones available, e.g., either `basic` or `simple`. The `basic` setting does not contemplate Perception Knowledge during the interpretation process. Conversely, the `simple` configuration relies on perceptual information, enabling a context-sensitive interpretation of the command at the predicate level.

During the initialization, a specific output format can be chosen, among the available ones. `xdg` is the default output format, where the interpretation is given in the eXtended Dependency Graph (XDG) format, an XML compliant container (see [9]). In the `amr` format, the interpretation is provided as Abstract Meaning Representation (AMR)Abstract Meaning Representation (AMR) (see [7]). Finally, `cfr` (CFR) is a format for the predicates produced by the chain defined in [146], in the context of RoCKIn competition. The language parameter allows to choose the operating language of LU4R. At the moment, only `en` (English) and `it` (Italian) versions are supported.

Service phase. Once the service has been initialized, it is possible to start asking for interpreting user utterances. The server thus waits for messages carrying the utterance transcriptions to be parsed and entities of the Semantic Map. Each sentence here corresponds to a speech recognition hypothesis. Hence, it can be paired with the corresponding transcription confidence score, useful in the Re-Ranking phase. The body of the message must then contain the list of hypotheses encoded as a

JSON array (Listing 5.2), called `hypotheses`, where each entry is a `transcription` paired with a `confidence`, according to the following syntax:

```
{
  "hypotheses":[
    {
      "transcription":"take the mug next to the keyboard",
      "confidence":"0.9",
      "rank":"1"
    },
    ...
    {
      "transcription":"take them all next to do keyboard",
      "confidence":"0.2",
      "rank":"5"
    }
  ]
}
```

Listing 5.2. JSON string of a list of transcriptions

The JSON must be passed as `hypo` parameter of the HTTP request. The `rank` attribute is redundant and it used only as an additional check.

Furthermore, when the `simple` configuration is selected, the input *can* include the list of `entities` populating the environment the robot is operating into (i.e., the set $\mathcal{E}_{\mathcal{PPK}}$ defined in Section 5.2.2), again encoded as a JSON array (Listing 5.3). Despite of the representation of the environment adopted by the robot, this environment-dependent interpretation process requires the following information for each `entity` *perceived* by the robot, i.e, collected into the Semantic Map, in this setting:

- the `type` of each entity; it reflects the class to which each specific entity belongs (e.g., it is an object, such as a `Cup`, `Keyboard`, or a location, such as `Living_Room` or `Kitchen`);
- the `preferredLexicalReference` used to refer to a class of objects; it is crucial in order to enable a linguistic grounding between the commands uttered by the user and the entities within the environment. These labels are expected to be provided by the engineer initializing the robot. For example, an entity of the class `Cup` can be referred by the string *mug*. If no label is given, it is derived by the name of the corresponding class, so that *cup* can be used to refer to the objects of the class `Cup`;
- in the case the engineer provides more than one label, these can be specified through `alternativeLexicalReference`, as a list of alternative naming for a given entity;
- the position of each entity is essential to determine the shallow spatial relations between entities (e.g., two objects are `near` or `far` from each other). To this end, each entity is associated with its corresponding `coordinate` in the world, in terms of planar coordinates (`x`,`y`), elevation (`z`) and `angle` as the orientation. At this moment, a simple grid map approach is used. Two objects are considered *near* whenever their Euclidean distance is less than 2, *far* otherwise.

All the above information can be provided to LU4R through the following JSON input:

```
{
  "entities":[
    {
      "atom":"p1",
      "type":"Cup",
      "preferredLexicalReference":"cup",
      "alternativeLexicalReferences":["cup","mug",...],
      "coordinate":{
        "x":"13.0",
        "y":"6.5",
        "z":"3.5",
        "angle":"3.5"
      }
    },
    {
      "atom":"k1",
      "type":"Keyboard",
      "preferredLexicalReference":"keyboard",
      "alternativeLexicalReferences":["keyboard","console",...],
      "coordinate":{
        "x":"12.0",
        "y":"8.5",
        "z":"0.0",
        "angle":"1.6"
      }
    }
  ]
}
```

Listing 5.3. JSON string of a the entity list

The above JSON string must be passed as `entities` parameter of the HTTP request.

Finally, the service can be invoked with a HTTP POST request that puts together the `hypo` and `entities` JSONs as follows:

```
http://IP_ADDRESS:PORT/service/nlu
POST parameters: hypo={"hypotheses":[...]}
                 entities={"entities":[...]}
```

5.6. Contributions

This chapter presented a comprehensive framework for the definition of robust natural language interfaces for HRI, specifically designed for the automatic interpretation of spoken commands towards robots in domestic environments. The proposed solution allows to inject **contextual evidence** into the interpretation process. It relies on Frame Semantics and supports a structured learning approach to language processing able to produce meaningful commands from individual sentence transcriptions. A hybrid discriminative-generative learning method is proposed to map the interpretation process into a cascade of sentence annotation tasks.

Starting from [17, 166, 170], this thesis defines a systematic approach to enriching the example representation with additional feature spaces not directly addressable

by the linguistic level. The aim is to leverage knowledge derived from a semantically-enriched implementation of a robot map (i.e., its Semantic Map), expressing information about the existence and position of entities surrounding the robot, along with their semantic properties. Observations extracted from the Semantic Map and useful for the interpretation are then expressed through a feature modeling process. Thanks to the discriminative nature of the adopted learning mechanism, such features have been injected directly into the algorithm. As a result, command interpretation is made dependent on the robot's perception of the environment.

The proposed machine learning processes have been trained by using an extended version of HuRIC. This corpus, originally composed of examples in English, has been improved by collecting also a subset of examples in Italian. Moreover, each example has been paired with the corresponding Semantic Map, linking the command to the environment in which it has been uttered and enabling the extraction of valuable contextual features. This novel corpus promotes the development of the proposed interpreting cascade in more languages but, most of all, it will support the research in grounded natural language interfaces for robots.

The empirical results obtained over both languages are quite impressive, specially when the system is evaluated in a real scenario (end-to-end cascade evaluated against the semantic head). The results suggest several observation. First, they confirm the effectiveness of the proposed processing chain. In fact, even when only linguistic information are exploited, the system obtains interesting results. Second, they prove the effect of contextual features extracted from the Semantic Map, which contributed, with different extent, to the improvement of each sub-task. Finally, the results promote the application of the same approach in different languages. In fact, the systematic extraction of both linguistic and contextual features makes the system easy to be extended to other languages.

In conclusion, the contributions of this chapter are: (i) the introduction of a grounded language understanding systems for the interpretation of robotic commands, (ii) the systematic injection of contextual features into the learning/tagging system, that allows to interpret NL commands coherently with the operational environment and the targeted robotic platform, (iii) experimental evaluations of the proposed models, that emphasize the impact of contextual information with respect to the addressed task, (iv) a corpus of robotic commands (HuRIC) that can support the development of data-driven systems for the interpretation of human language, and (v) a ready-to-use tool (LU4R) that can be deployed onto any robotic platform, and that implements the paradigms proposed in this chapter.

All the above resources and findings make a step forward towards the development of robots like Roy. In fact, when dealing with situated Spoken Human-Robot Interaction (SHRI), the pure linguistic dimension is not enough to properly represent the language meaning; on the contrary, we proved that even the process of understanding a command can be enhanced by leveraging the information provided by the operational context.

Chapter 6

The Role of Context in Dialogue Modeling

This chapter focuses on the problem of learning dialogue policies to efficiently support robot teaching tasks (see Section 1.2.3) so that the effort required by humans in providing knowledge to the robot through dialogue is minimized. This is a feature that Roy exhibits (Section 1.1); in fact, while learning the objects of the environment, it is able to leverage the acquired information to autonomously infer new knowledge. The underlying idea of the proposed approach introduced in [173], is that **perceived context**, acquired through robot's sensors, can represent a valuable source of information to drive the teaching process (Figure 6.1). The addressed task refers to the process of acquiring semantic attributes through natural language interactions that is directly related to Human Augmented Mapping (HAM) (presented in Section 2.5.1). In fact, the proposed interactive system for HAM enables the acquisition of semantic properties of objects through interactions that become more and more efficient over time, as the tutoring effort is quickly minimized. The dialogue policy is based on a multi-objective Markov Decision Process (MDP), while the optimization problem is solved through Reinforcement Learning (RL). To this end, such cost-effective teaching can be obtained by exploiting the increasing reliability of the visual classifiers that are learned incrementally over the perceived context. The visual classifier is here realized by combining a pre-trained Convolutional Neural Network (CNN) with a Load-Balancing Self-Organizing Incremental Neural Network (LB-SOINN). Images acquired by the robot's depth camera are preprocessed and fed to the CNN, which maps them onto a lower-dimensional feature space. With every new input feature vector, the LB-SOINN is able to adapt in order to reflect the underlying topology of the data distribution. The operational hypothesis is thus that, whenever perceptual information of the operational context is properly exploited, the Dialogue Manager (DM) is able to minimize the tutoring cost, resulting in a less tedious interactive mapping process.

Such hypothesis has been validated by running several simulated empirical investigations, whose outcomes show that the resulting adaptive dialogue strategy is able to find an optimal trade-off between the classifier accuracy and the tutoring cost. Moreover, results are encouraging for the deployment of the system in real scenarios. In fact, the experiments showed that the policies can be successfully trained on a

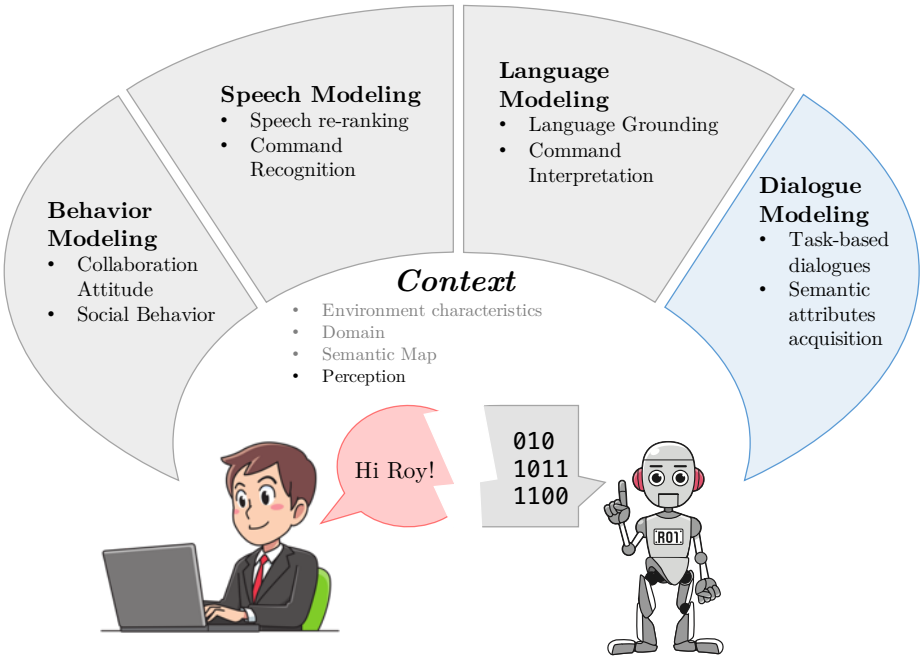


Figure 6.1. Task-based dialogic interactions are more effective when context is properly exploited.

small set of examples and yet, generalize well to perform properly on larger datasets.

The chapter is thus structured as follows. Section 6.1 presents the specific literature and contextualizes the work, while in Section 6.2 the overall framework is presented. A quantitative evaluation of the system is provided in Section 6.3, along with a discussion of the results (Section 6.4). Section 6.5 shows a demonstration of the framework in a real scenario. Finally, Section 6.6 reports the conclusions of the work and the contributions of the chapter.

6.1. Related Work

In recent years, several systems have aimed at mapping the operational environment of a robot with semantic attachments. According to the definition provided in [126], the resulting Semantic Map is a representation of the environment that couples the spatial structure with semantic information concerning locations and objects therein. In this respect, such a process is often carried out by associating symbols to physical elements of the environment [77].

Several works treat the problem as a fully automated process. In [28] the authors focused on the recognition of rooms by extracting *valuable* attributes. In [27, 60, 68, 120] topological maps are built upon the metric ones, enabling the robot to perform an *aware* navigation of the environment. With the recent advances in object recognition and categorization, several approaches exploiting visual features have been proposed [121, 181].

A few approaches rely on the presence of a human in the learning loop (HAM), acting as a tutor who instructs the robot to learn the environment. In fact, fully

automated semantic mapping systems are error-prone and do not provide the wide-range knowledge that can be acquired by interacting with a user through speech. For example, in [123] the authors rely on a multivariate probabilistic model to associate semantic labels to spatial regions and on the support of the user in selecting the correct one. Conversely, in [95] clarification dialogues are used to support the mapping process. Such an approach is further extended in [187] to create conceptual representations of indoor environments which are used in human-robot dialogue. In [130] the authors use manifold modalities for a multi-layered semantic mapping system to categorize places and build topological maps. More recently, in [67] a human-augmented semantic mapping system is presented. The authors focus on an online setting, where the semantics of objects is acquired incrementally through long-term interactions with the user. The dialogue policy is implemented beforehand through Petri Net Plans (PNPs) and the robot is not enabled to infer semantic properties of new objects from the acquired knowledge. In [163], the authors propose an approach for the opportunistic acquisition of objects descriptors (or attributes) relying on the users' feedback. Though the task is similar, our system focuses mainly on the minimization of the tutoring cost during the teaching activity.

There are several novelties that differentiate this approach from the literature of the field. First, the proposed interactive system obsessively exploits contextual information through an incremental object classifier to collect visual evidence of the objects. Such knowledge is then used to automatically recognize new objects, supporting a quick acquisition of the semantic map. The dialogue interactions benefit from an analysis of the visual classifier reliability, as this information is exploited to determine whether to ask the human tutor a clarification question or not. Moreover, the proposed DM driving the semantic attribute learning is entirely data-driven. This feature is essential to enable the deployment and optimization of a robotic platform in heterogeneous environments, interacting with different users speaking different languages. Finally, the acquisition of the policies can be performed on a very small set of examples, while still showing good performance when tested on larger datasets. This feature enables the deployment of our approach in a long-term mapping scenario. Moreover, the policies may be further improved while the system is operating and, hence, adapted to the specific user.

6.2. Acquiring Semantic Attributes through Interaction and Perception

Semantic Mapping is a task that involves several sub-problems, such as the representation of the semantic properties, route planning, interaction management, and sensing ([171]). In this chapter, we will focus on two of them: (i) the management of the dialogue for the acquisition of semantic properties, and (ii) the memorization of synthetic representations of the object that are used to compose the semantic map. Table 6.1 shows two running examples of possible interactions, where tutor and learner interactively exchange information about the category of a particular visual object. On the left, the interaction is triggered by the user, that asks the robot about the category of a specific object; on the right, the robot takes the initiative and the user teaches a new object by replying to its questions. It is worth emphasizing that

(a) Tutor Initiative	(b) Learner Initiative
T(utor): what is this object?	**L**: a shampoo bottle, am I right?
L(earner): I have no idea.	**T**: no, it is not.
T: a shampoo bottle.	**L**: so what is it?
L: okay, shampoo bottle.	**T**: an apple.
T: good job.	**L**: okay, got it.

Table 6.1. Dialogue Examples from the synthetic Dialogue Collection: *(a) the user takes the initiative (b) the learner takes the initiative.*

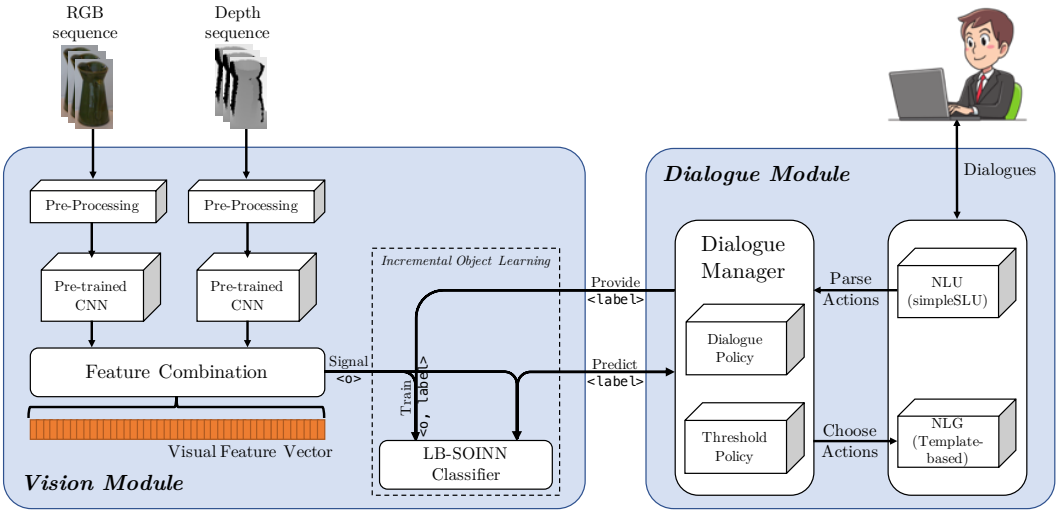


Figure 6.2. Overview of system architecture for semantic attributes learning

at the very first interaction, the robot tries to make a guess about the category of the object (i.e., *"a shampoo bottle, am I right?"*). This is exactly the behavior we might want, a system that is able to exploit the acquired knowledge to improve its capabilities. In fact, even though in this interaction the category prediction is wrong, a better visual classifier would be able to catch the correct label and the interaction would be composed of just two dialogue turns (e.g., **L**: *"a shampoo bottle, am I right?"* – **T**: *"yes, it is."*).

In this respect, we describe hereafter the proposed interactive multi-modal system in support of learning semantic map attributes (e.g., visual classes) through natural conversational interaction with human tutors. RL is exploited for the learner's dialogue strategy optimization, while the tutor is simulated here in a data-driven fashion using synthetic dialogue examples (e.g., Table 6.1). The same method can be applied to a real scenario, where users interact with the deployed system about the objects present into the environment.

6.2.1. Overall System Architecture

In this section, the proposed system architecture (see Figure 6.2) is introduced. It is composed of two essential modules: a *Vision Module* and a *Dialogue Module*, that actively interplay to achieve the goal of acquiring the object label.

Vision Module. The *Vision Module* is built upon the architecture proposed in [128], which accomplishes incremental online learning by combining a LB-SOINN [188] and a pre-trained CNN based on the architecture proposed in [94]. The combination of these two algorithms allows to leverage the great representational power of deep CNNs while retaining the ability to adapt to new data points incrementally provided by the LB-SOINN. In particular, the latter is an essential requirement for robots operating in real and dynamic environments, since it is impossible to anticipate, synthesize and design all the possible situations that the robot may encounter during its operation.

The system operates on two channels, processing RGB and depth images respectively. Both channels resize and rescale the input images to the format expected by the CNN. The depth channel further processes the depth image to produce a colorized surface normals image. Once the images have been preprocessed, they are fed into two identical pre-trained CNNs, that output the corresponding feature vectors. These feature vectors are further combined by computing their average; the result is finally used to update and grow the LB-SOINN. Effectively, this module allows to ground noun words such as *"apple"* and *"shampoo"*, which are used as parameters of the Dialogue Acts in the dialogue module, onto their visual representations.

Dialogue Module. This module relies on a classical architecture for dialogue systems, composed of Dialogue Manager (DM) and Natural Language Understanding (NLU), as well as Natural Language Generation (NLG) components. These components interact via Dialogue Act representations [156] (e.g., $inform(obj = apple)$, $ask(object)$). The NLU component processes human tutor utterances by extracting a sequence of key patterns, slots and values, and then transforming them into dialogue-act representations, following a list of hand-crafted rules. The NLG component makes use of a template-based approach that chooses a suitable learner utterance for a specific dialogue act, according to the statistical distribution of utterance templates from synthetic dialogue examples. Finally, the DM component is implemented with an optimized learning policy using RL (see Section 6.2.3). This optimized policy is trained to (i) process Natural Language (NL) conversations with human partners, and (ii) achieve a better balance between classification performance and the cost of the dialogue to the tutor in an interactive learning process.

6.2.2. Visual Object Classification

As mentioned in Section 6.2.1, one of the main components of the applied vision module is the LB-SOINN (for more details, see Appendix A.2.1). This method is based on the Self-Organizing Map [87] and is able to learn the underlying topology of the data distribution, without the need to specify the number of classes in advance. Operationally, each node in the network has an associated weight which lives in the feature space. Every time a new image is input to the vision module, the LB-SOINN algorithm assesses whether a new node has to be added to the network, based on the feature vector similarity to all the other nodes' associated weights. If no node is added, then the closest node and its neighbors weights are updated, and the two closest nodes are joined by an edge. In this way, the structure of the network evolves to reflect how the data is distributed in the feature space.

As the focus of this task is on object classification as opposed to image classification, the contributions of all the images corresponding to the same object have been considered in order to produce a classification result. However, this procedure is consistent with a real scenario, where a robot may look at the same object from different views and infer what it is, based on the consensus achieved from the results for every individual view.

In order to classify each image, a confidence score is computed as the average inverse distance between its feature vector and the weights of every node belonging to a given class, i.e., the closest the feature vector is to the nodes corresponding to the given class, the bigger the score that class will receive. This procedure is repeated for every class in the network and then the resulting confidence scores are normalized, such that they resemble probabilities, i.e., their sum is equal to 1. Hence, the class that receives the highest normalized score is chosen as the classification result for that image. In particular, the normalized confidence score is computed as follows:

$$conf = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{n_i}\sum_{j=1}^{n_i}\frac{1}{D(p,q_j)}\right) \tag{6.1}$$

where n_i is the number of nodes for the i^{th} class, N is the number of classes and $D(p,q_j)$ is the distance between the feature vector p and the weight q_j corresponding to the j^{th} node. The procedure proposed by [188] is applied for the computation of $D(p,q_j)$, as a combination of Euclidean and cosine distances affected by a weight, that is, in turn, a function of the dimensionality of the feature vector, i.e., for low-dimensional features, the Euclidean distance will be dominant whereas for high-dimensional features, the cosine distance will be dominant:

$$\begin{aligned} D(p,q_j) = \frac{1}{\eta^d}\frac{EU_{pq_j} - EU_{min}}{1 + EU_{max} - EU_{min}} + \\ + \left(1 - \frac{1}{\eta^d}\right)\frac{CO_{pq_j} - CO_{min}}{1 + CO_{max} - CO_{min}} \end{aligned} \tag{6.2}$$

where d is the dimension of the feature vector, $\eta = 1.001$ is a pre-defined parameter, EU_{pq_j} is the Euclidean distance between p and q_j, EU_{max} and EU_{min} are the maximum and minimum Euclidean distances between any two nodes in the network respectively, and CO_{pq_j}, CO_{max} and CO_{min} are the equivalent quantities corresponding to the cosine distance measure.

Finally, a voting schema is adopted over all the images of the object, normalizing again for all the classes that were returned as potential candidates. The final result of an object classification is defined as the class that obtains the highest consensus among all the images, i.e., highest *probability*.

6.2.3. An Adaptive Dialogue Strategy for Interactive Mapping Tasks

For learning the dialogue policy, we foster the idea that a comprehensive teachable system should learn as autonomously as possible, rather than involving the human tutor too frequently [152]. Accordingly, as pointed out in [184], an intelligent agent should provide the capability of finding an optimized trade-off between the goal

achievement and the tutoring cost in a particular task. In other words, given the visual mapping task, the agent should be able to learn the scene accurately, and with little effort from human instructors. To this end, in order to optimize the trade-off, the interactive mapping problem can be formulated into two sub-tasks: *when* and *how* to learn the grounding mappings. These sub-tasks are trained using RL with a multi-objective MDP, consisting of two sub-MDPs (for more details, see Appendix A.1.2). The robot behavior is characterized by the following sequence of steps: (i) a visual instance is shown to the agent/learner; (ii) based on the outcome of the instance classification, i.e., a confidence score for each category acquired so far, the agent/learner determines *when* and *how* to ask questions; (iii) the dialogue continues with a response from the user.

When to Learn

In the first MDP, called *adaptive-threshold MDP*, the policy is required to learn when the robot needs to acquire useful information from human tutors (i.e., objects' labels), where a form of active learning is taking place: the agent learns to ask questions about particular objects only whenever it is uncertain about its own predictions. In order to establish the visual classifier reliability, a positive confidence threshold is adopted, which determines when the learner can trust its visual predictions. This threshold represents the core role in achieving an optimal trade-off between the classification performance and the tutoring cost, since the learner's behavior (e.g., whether to seek feedback from the tutor or not) is dependent on this threshold.

To this end, the agent acquires here an adaptive strategy that aims at maximizing the overall performance by properly adjusting the confidence threshold in the range from 0.9 to 1. Moreover, each training episode terminates when the agent passes through all instances in the visual dataset. The problem is thus modeled as follows.

State Space. The *adaptive-threshold MDP* operates on a 2D state space, consisting of `curThreshold` and `levelRel`. While `curThreshold` represents the positive threshold that the agent is currently applying, `levelRel` is applied to locally measure the reliability of the visual classifier after a single learning step. To this end, in order to define a learning step, the total number of instances (objects) is clustered into bins, with each bin containing n_B instances and representing a single learning step. Hence, `levelRel` is computed as a discretization of the Local Accuracy into three levels as below:

$$\texttt{levelRel} = \begin{cases} 1, & \text{if } Acc_{loc}^{[-1,1]} > 0 \\ 0, & \text{else if } Acc_{loc}^{[-1,1]} = 0 \\ -1, & \text{otherwise} \end{cases} \tag{6.3}$$

where $Acc_{loc}^{[-1,1]}$ represents the *Local Accuracy* Acc_{loc} of classifiers (defined in Section 6.3.1), rescaled between -1 to 1. In practice, *Local Accuracy* is expected to provide a quantitative extent of the visual classifier performance over a single individual bin of objects.

Action Selection. Based on the performance of the classifier on the previous learning step, the agent updates the state space of the MDP by either increasing/de-

creasing the current confidence threshold by 0.02 or keeping it to the same value. Hence, the possible actions this agent may choose are INCREASE, DECREASE and KEEP-THE-SAME.

Reward Function. The local reward function R_{loc} for learning the adaptive-threshold agent is defined to be proportional to the Local Accuracy Acc_{loc} of the visual classifier, computed over each bin of n_B instances. Hence, the system will reward the action if the rescaled accuracy $Acc_{loc}^{[-1,1]}$ is greater than 0, otherwise, the action is penalized.

How to Learn Using Dialogue

The second MDP aims at effectively acquiring useful information through interactions with human partners. In this case, the action selection is highly biased by the previous MDP, as it depends on the threshold level controlled by the adaptive-threshold MDP. In fact, if the learner has a low confidence on its predictions, i.e., the confidence score output by the visual classifier is lower than 0.5, it may ask WH-questions to directly acquire correct labels from the tutor (e.g., *"what is this object?"*). Otherwise the learner will make a guess about the label by either asking a YES-NO-question (e.g., *"is this an apple?"*), or directly assigning the predicted label to the object, thus without relying on the user's intervention (e.g., *"this is an apple."*). In addition, the learner is also expected to produce coherent conversations with a human partner, i.e., understand particular dialogue intents from humans and properly produce the next responses. In this MDP, every single dialogue represents an episode and is terminated when the class name is either taught by a human tutor or inferred through a sufficiently high confidence score. Accordingly, the RL process and the corresponding MDP have been configured as follows.

State Space. The dialogue policy initializes a 3D state space, defined through the variables `cStatus`, `preDAts` and `preContext`. In particular, `cStatus` is applied to represent the current status of predictions about a particular object and evaluated as follows:

$$\texttt{cStatus} = \begin{cases} 2, & \text{if } \texttt{conf} > \texttt{curThreshold} \\ 1, & \text{else if } 0.5 \geq \texttt{conf} \geq \texttt{curThreshold} \\ 0, & \text{otherwise} \end{cases} \tag{6.4}$$

where `conf` is the confidence score about a specific object and output by the visual classifier and `curThreshold` is the threshold level computed by the adaptive-threshold MDP. Hence, the status level is a function of the confidence score and the positive threshold described above. Conversely, `preDAts` represents the actions the tutor performed in the previous dialogue turn. In fact, this variable is meant to capture the short history of the interaction flow, represented by the previous dialogue turn. This variable is essential to properly make the robot turn to be dependent on and coherent with what the user said. Finally, `preContext` represents whether a visual category was mentioned in the dialogue history and what category it is (e.g. class of the object, its color, shape, ...). In this work, as only the class name of the

visual object is taken into account, `preContext` contains one out of two values, i.e., unmentioned (`U`) and object class name (`C`).

Action Selection. The actions are chosen based on the analysis of task-oriented dialogue actions occurring in a dataset of hand-crafted dialogue examples (see, for example, Table 6.1), where each sentence is labeled with its corresponding dialogue acts. This set includes dialogue acts like Wh-questions, Polar-questions, DoNotKnow, Acknowledgment, as well as Listening.

Reward signal. The reward signal is defined to be a global function R_{glob}:

$$R_{glob} = 10 - Cost - Penal \tag{6.5}$$

which takes into account the cumulative cost of the tutoring process (*Cost*, defined in Section 6.3.1) in a single conversation, and penalties (*Penal*) for inappropriate actions performed by the learner (e.g., if the learner does not answer a question).

6.3. Experimental Evaluation

The experimental setup aims at simulating a semantic mapping task, where the robot navigates throughout the environment to acquire semantic properties of the objects populating its world. Notice that while the problems of planning and navigation are out of the scope of this work, the focus here is on the *category* (or label) of objects (e.g., `apple`, `calculator`, ...). However, it worth emphasizing that the approach can be easily extended for the acquisition of other properties, such as *color* and *shape*.

Figure 6.3 shows the GUI used to visualize the simulated environment. The robot keeps moving into the assigned area (grid map in the center of Figure 6.3), seeking for unknown objects (the *red* squares in the grid). Once an item is reached (orange cells), the visual classifier is fed with images corresponding to the current instance (e.g., on the bottom right box in Figure 6.3). The confidence score provided by the visual classifier is then used for deciding whether to assign the predicted label (without an interaction with the user) or to ask a Polar/Wh-question, according to the current threshold level. At the end of the interaction, the object is finally labeled (green cells) with the category provided by the user[1], and the corresponding images are used to train the visual classifier. It is worth noting that the classifier is updated only whenever the label is provided by the user. That is, when the agent/learner trusts the classifier, we assume that the specific instance is already well represented in the model. Such a conservative approach aims at avoiding possible noise introduced into the net when unnecessary images are learned.

6.3.1. Evaluation Metrics

The evaluation of the trained learning/dialogue strategy is based on metrics inspired by the PARADISE evaluation framework [176] for task-oriented dialogue systems.

[1] In this work, lexical variation is not taken into account; instead, categories are identified through a fixed vocabulary.

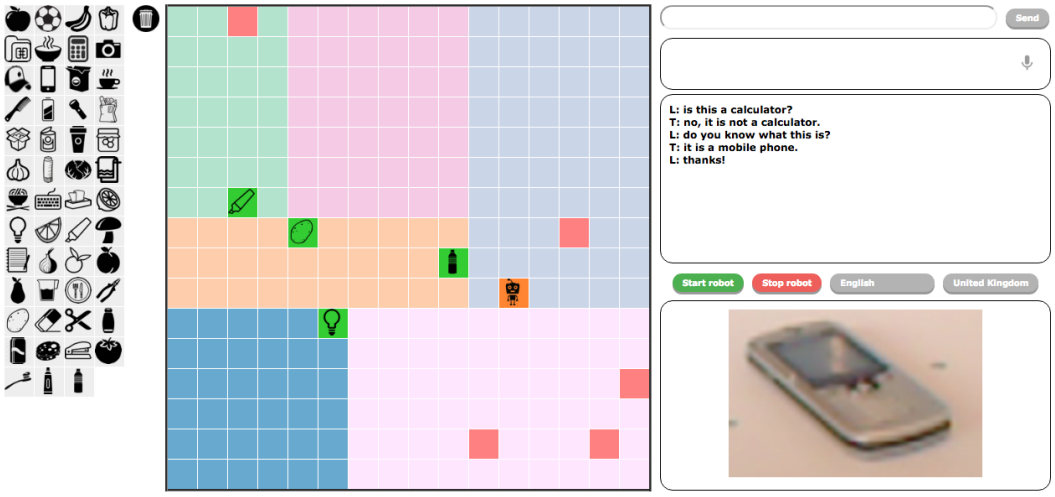


Figure 6.3. The simulated environment for interactive semantic attributes acquisition. The *left* block shows the labels available within the dataset; the grid map in the *center* emulates the environment in which the robot is moving, where green cells refer to correctly recognized objects, red cells are the objects that have not been already discovered, while the orange cell is the targeted object; on the *right*, the dialogue flow and the images of the targeted object are shown

Hence, the overall performance of the policy executed by the robot takes into account both the task success (classification accuracy) and cost (the tutoring effort within dialogues). As the ultimate goal of the work is the minimization of the tutoring cost, the cost will deserve a special attention when analyzing the results. However, the system in the experiment is required to achieve and retain a better trade-off between the accuracy and the cost through an interactive learning period. More details about these metrics are described below.

Local Accuracy. The *Local Accuracy* is the metrics used to evaluate the performance of the visual classifier and, in turn, to assess its reliability in predicting objects' category. To this end, the learning performance of the classifier has been measured using visual instances which may have been seen in previous learning steps. Hence, the system is able to self-test on objects that it has seen before, as its learning progresses. To this respect, the *Local Accuracy* (Acc_{loc}) of the i-th bin is computed at the end of the bin using the initial predictions obtained for each instance during the processing of the bin. It is worth noting that such procedure is applicable only whenever online learning schemes are applied, as it measures the performance of the partial models, that are not available in batch learning.

However, even though such a procedure might result counterintuitive, it represents one of the novelties and deserves a more detailed explanation. Operationally, as sketched in Figure 6.4, for each instance in the dataset, the prediction from the visual classifier is obtained and if the prediction is correct, the True Positives (TP) are increased by 1, otherwise the instance is learned. When the n_B instances of the

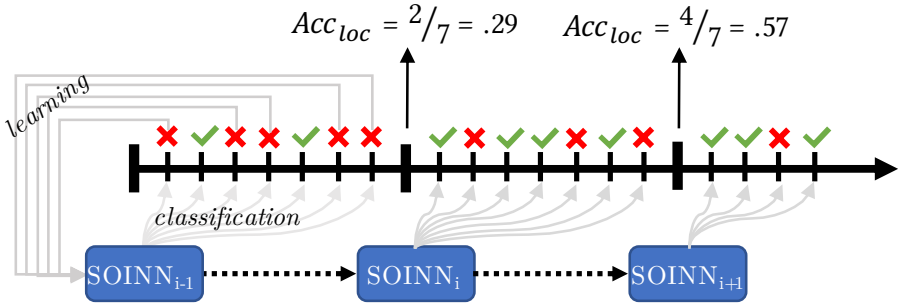


Figure 6.4. Local Accuracy evaluation

Action	C_{inf}	$C_{ack/reject}$	C_{crt}
Cost	5/0	0.5	5/0

Table 6.2. Table of Costs to the Human tutor within Conversation

bin have been processed, the *Local Accuracy* is evaluated as follows

$$Acc_{loc} = \frac{TP}{n_B} \tag{6.6}$$

and the TP value is reset. In this way, the evaluation score obtained after each bin is not biased by the training data. In fact, the prediction of each object is made with the model acquired so far and the object is learned only if the prediction is wrong.

Cumulative Tutoring Cost. As already outlined by the PARADISE framework proposed in [176], the performance of a dialogue system is also a function of a combination of cost measures. Intuitively, cost measures are calculated on the basis of any user or agent dialogue turns. Skočaj et al. [152] pointed out that a comprehensive system should be able to learn as autonomously as possible, rather than involving the human tutor too frequently.

Hence, the Cumulative Tutoring Cost (or simply *Cost*) is applied to reflect the effort needed by a human **tutor** in interacting with the system/robot. In literature, a wide range of cost measures have been proposed and exploited. Given the learning task, there are four possible costs that the tutor might incur in, as defined below (and summarized in Table 6.2):

- C_{inf} (*Inform*) refers to the cost of the tutor providing information on the name of the specific visual instance (e.g., *"this is a shampoo bottle"*); it may be either 5 or 0, depending on whether the dialogue act is present or not within the sentence;

- C_{ack} (*Acknowledgment*) is the cost for a simple confirmation (like *"yes"*, *"right"*); it is set to be 0.5;

- C_{reject} (*Rejection*) is the cost for a simple rejection (such as *"no"*, *"it is wrong"*); it is set to be 0.5;

- C_{crt} (*Correction*) is the cost of correction of a statement/polar question (e.g., *"no, it is an apple"*); it is also set to be either 5 or 0.

The cumulative cost is evaluated as sums of these action-costs across all dialogues

$$Cost = \sum_{i=0} C^i_{inf} + \sum_{j=0} C^j_{ack} + \sum_{k=0} C^k_{reject} + \sum_{l=0} C^l_{crt} \tag{6.7}$$

where a single dialogue is considered as the interaction required to acquire a single visual instance.

6.3.2. Visual Object Dataset

The proposed system has been evaluated over the Washington RGB-D Object Dataset [99]. This dataset is acquired using a Kinect-like sensor and consists of 300 household objects organized into 51 categories. For each object, video sequences of full 360° rotations at three different heights of the sensor are available. In addition, the dataset is provided with cropped versions of the objects and binary masks which aid in pre-processing the images. However, cropping and segmentation are outside the scope of this work and are assumed to be available in a full system implemented on a robotic platform.

The particular nature of this dataset, i.e., the sensor used and the acquisition setup, allows to simulate a real interactive scenario, where a robot is able to obtain different sequential views of the same object. Moreover, in order to reduce randomness, the size of the dataset has been reduced by considering only 120 random images per object. Therefore, the models are trained on a random subset that accounted for 50% of the images and tested on a random subset that accounted for 25% of the images. This allowed us to speed up the learning process as well as to increase the degree of overlap between train and test subsets. Even though this overlap might result unfair in terms of performance evaluation, it is worth reminding that (i) the focus of this approach is not the visual classification itself, but the minimization of the tutoring cost, and (ii) this setup allows to reproduce a real situation, where the robot might encounter the same object more than once.

6.3.3. User Simulation for the Learning Task

In order to train and evaluate the dialogue agent, a user simulation was required; it resembles human behaviors on the task of teaching visual objects using a generic n-gram framework (see [186]). The simulated tutor is trained on a collection of synthetic dialogues (see dialogue examples in Table 6.1). Once the statistical distribution of the data is acquired, the user's action (e.g., INFORM, NEGOTIATION, REJECTS, ...) is predicted probabilistically. This simulation framework takes as input the sequence of N most recent words in the dialogue, as well as some optional additional conditions, and then outputs the next user response on multiple levels as required (e.g., full utterance, a sequence of dialogue actions, or even a sequence of single word outputs for incremental dialogue behavior). In this work, an action-based user

Dialogue Example (a)	Dialogue Example (b)
L: what is this object called? T: an apple L: okay, apple T: good job.	L: this is a shampoo, right? T: no, it is not shampoo, it is a cereal box. L: okay, got it.

Table 6.3. Example Conversations between the RL-based Learning Agent (L) and the Simulated User (T): *(a) Learner with low confidence (b) Learner with higher confidence.*

model was created, able to predict the next user response in a sequence of dialogue actions. The simulator then produces a full utterance by following the statistics of utterance templates for each predicted action.

6.4. Results and Discussion

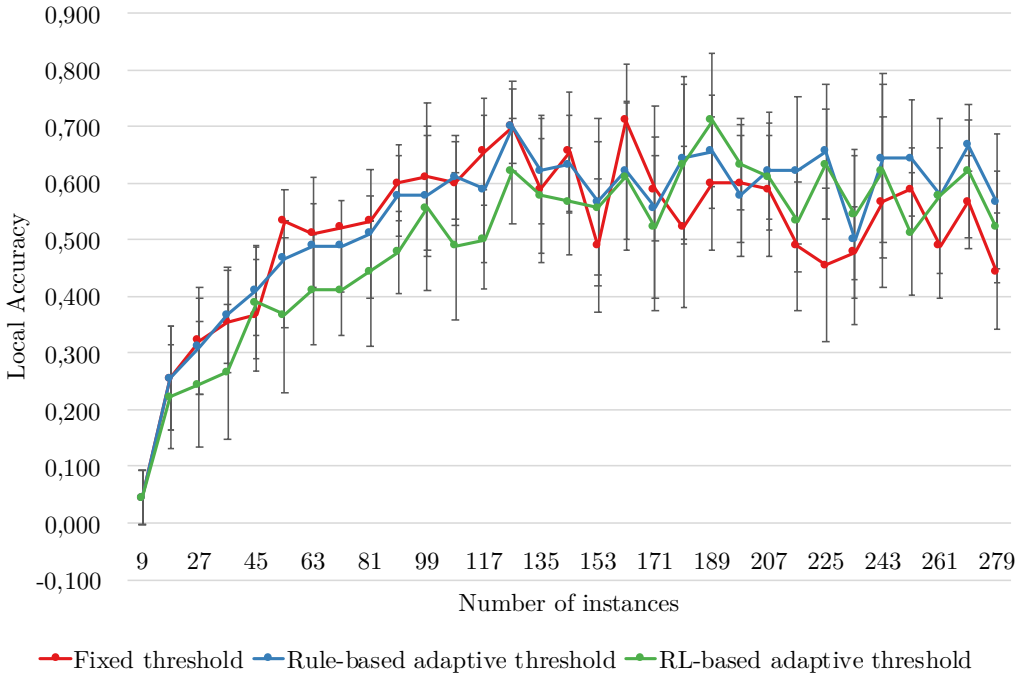
Several empirical evaluations aimed at determining the effectiveness of the adaptive-threshold MDP and the applicability of the approach in real scenarios.

The policies have been trained for 5000 episodes on a dataset of 48 instances, distributed over 10 classes randomly drawn from the Washington RGB-D Object Dataset. The instances were grouped in bins with $n_B = 8$ objects each. Table 6.3 provides exemplifying interactions between the learned RL agent and the simulated user, showing how the learner, in order to minimize the cost, favors to take the initiative. The Burlap RL library [109] has been adopted to model the MDPs and learn the policies. In particular, the State-Action–Reward-State-Action (SARSA) algorithm [157] is used to learn the both the policies, with a greedy exploration rate of 0.1 and a discount factor of 1 (for more information about SARSA and its hyper-parameters, see Appendix A.1.2).

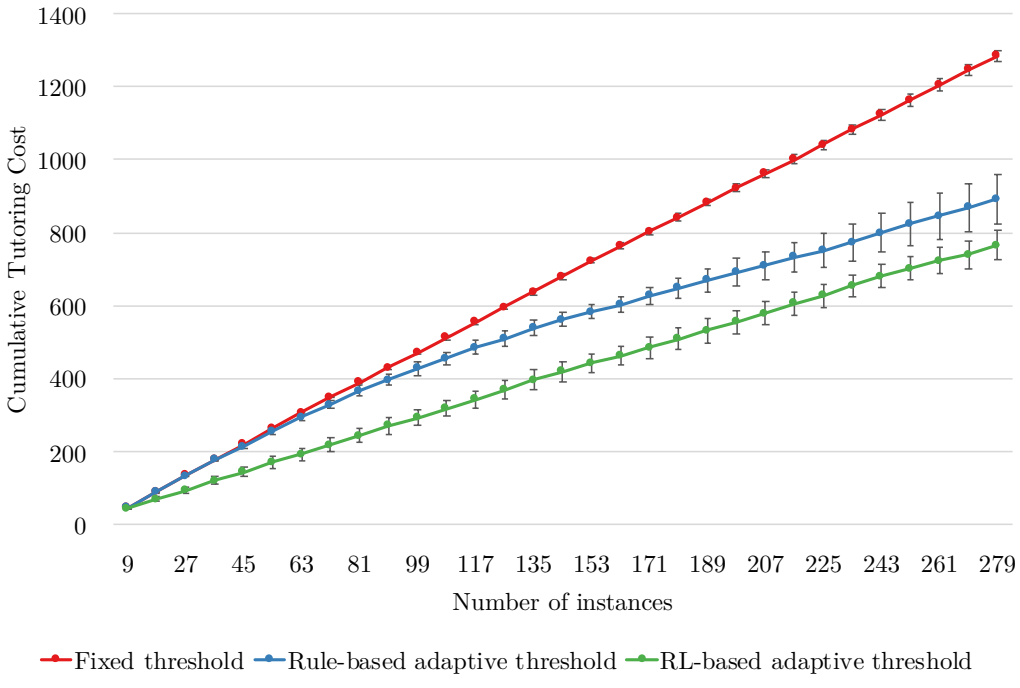
In order to prove the effectiveness of the policies over unseen objects, we tested them on a dataset of 25 classes (143 instances), where the overlap with the training set is minimal, repeating the experiment for 10 folds. The size of the bins was $n_B = 9$. In our scenario, the robot keeps navigating the environment, so after a while, it may reach an object that it has already seen before. To simulate this, we replicated the number of instances by 2, randomly shuffling the dataset, both when training and testing the policies. Hence, for example, the instance `apple_2` (belonging to the `apple` category) will be processed twice.

In Figure 6.5 the plots obtained from the experiments are reported. Results are provided in terms of *Local Accuracy* (Figure 6.5(a)) and *Cumulative Tutoring Cost* (Figure 6.5(b)). In such analysis, three different approaches for adjusting the confidence threshold have been compared.

The first setting applies a *Fixed threshold* (FT) set to 1. This is the baseline, where the robot keeps asking questions, as the classifier outcomes are always less than (or equal) to 1. The second setting relies on a *Rule-based adaptive threshold* policy (RT) to adjust the threshold. This hand-crafted policy modifies the threshold as follows: whenever the ΔAcc_{loc} is positive, the threshold is decreased by 0.02; conversely, if the ΔAcc_{loc} is negative, it is increased by 0.02; otherwise, it is not modified. Finally, the proposed policy, acquired through the approach described so



(a) Local Accuracy



(b) Cumulative Tutoring Cost

Figure 6.5. Results of the experimental evaluation, provided in terms of *Local Accuracy* (left) and *Cumulative Tutoring Cost* (right), along with 95% Confidence Intervals.

far (*RL-based adaptive threshold*, or *RLT*), has been evaluated.

ANOVA test is performed to evaluate the significance of the different settings for Acc_{loc} and *Cost*. The outcomes suggest that there are no significant differences in the local accuracy under the three different threshold conditions. However, this is not true for the *Cost*, where the p-value is $p < 1 \times 10^{-14}$. The ANOVA results are confirmed by a post-hoc pairwise comparison over the *Cost*, performed through t-tests. The outcomes show that the *RLT* policy has significantly less tutoring cost than the others, namely the *FT* ($p < 4 \times 10^{-14}$) and the *RT* policies ($p < 0.006$).

As expected, in the first setting, the *Cost* is represented by a straight line, as the learner applies the same dialogue pattern for all the interactions. Since the user always provides a label for the given object, it is plausible to expect a better Acc_{loc} curve. Instead, it seems that this metric is affected by *noise*: even though the classes are well represented within the LB-SOINN model, the learner keeps updating the network by injecting unnecessary examples. The *RT* policy seems to get acceptable results, as (i) the tutoring cost tends to decrease as more objects are processed, while (ii) the accuracy is not degraded. However, the *RLT* setting seems to outperform the other techniques. In fact, the *Cost* is always minimized and most importantly, starts to decrease from the very beginning of the process, i.e., the threshold is decreased as soon as the robot starts to trust the classifier. This behavior is essential, as the benefits of the RL-based threshold would be perceived even after a few interactions. At the same time, the Acc_{loc} curve seems to follow the same trend as the other settings, suggesting that the tutoring cost can be minimized without loss in accuracy. Nevertheless, once a considerable number of classes is acquired, the confidence values provided by the visual classifier are lower than in the early stages, due to a higher internal uncertainty of the network. Hence, even though the prediction for an instance is correct, but with a low confidence score, the threshold does not have the chance to decrease further since its lower bound is set to 90 (a conservative solution for the classifier trust). As a consequence, the *Cost* stops decreasing and the corresponding curve appears as a straight line.

6.5. Demonstration on Real Robot

In order to support the effectiveness of the proposed approach, the system has been deployed on a real robot for some preliminary tests. The targeted platform is a modified version of the Turtlebot 2 Robot[2] (Figure 6.6). While the base has not been modified, the structure on top is customized, in order to make the robot taller with respect to the off-the-shelf version. The robot is 107 cm high and features a tablet as an interface for the interactions. In fact, the Automatic Speech Recognition (ASR) module has been realized through the Google Speech APIs [78], available within the Android environment, in an ad-hoc mobile application. The robot has been equipped with the Asus Xtion Pro Live RGB-D camera. Though the nature of the resulting dataset is still the same as in the simulated scenario (for each shot, RGB and depth images are taken), the presence of a textured background and the hand holding the object might interfere with the learning process (segmentation and cropping of the object are outside the scope of the work). The robot was teleoperated by the

[2] `http://www.turtlebot.com/turtlebot2/`

Figure 6.6. The robot used in the real scenario demonstration

user. In fact, as it was not able to autonomously detect the presence of an object in front of the camera, it was forced to capture 30 RGB-D images on command by the user, i.e., by pressing a button on the joystick controller. Then, the pipeline proceeds as in the simulated experiment. Even though the performance has not been quantitatively measured, the system behaves as expected, minimizing the effort needed by the human in instructing the robot to acquire new objects. Hence, this further demonstration provides a preliminary evidence of the effectiveness of the proposed solution.

6.6. Contributions

This chapter focused on the problem of acquiring a dialogue policy to support interactive semantic mapping, with the goal of minimizing the users' tutoring cost. To this end, the project described in this section of this thesis proposed a multi-objective MDP Dialogue Manager, where the optimization problem is solved through RL and the interaction is made dependent on **contextual visual information**. In fact, while one MDP is devoted to the selection of the proper Dialogue Act, the other one modifies the level of trust in visual information. The latter is provided by an online visual classifier, based on a LB-SOINN. The benefits introduced by the adaptive threshold MDP have been evaluated through simulated empirical investigations that confirmed our initial hypothesis.

To sum up, the contributions of this chapter are: (i) the definition of a multi-objective RL framework for the acquisition of semantic attributes of objects populating the operating environment, (ii) the systematic exploitation of contextual visual information, encoded as images of the targeted object, through a comprehensive Machine Learning (ML) architecture that is able to provide guesses to the dialogue manager with the aim of minimizing the tutoring cost, (iii) the definition of a dedicated MDP, for controlling the reliability level of the visual classifier, and (iv) a

quantitative analysis of the impact of such contextual information in minimizing the tutoring cost.

The findings of this chapter are a further yet solid proof that perceivable context has a key role in situated Spoken Human-Robot Interaction (SHRI) tasks. In fact, through the use of such a knowledge, it is possible to design spoken DMs for robots like Roy, that are able to achieve their goal with a proper trade-off between autonomy and users' help.

Chapter 7

Conclusion and Discussion

This chapter concludes this dissertation and provides some remarks for the design and development of future robotic platforms interacting with humans in spoken language. In order to frame this thesis within the literature, in the following, a summary of each chapter is reported and the corresponding contributions highlighted.

7.1. Summary of Contributions

The goal of this thesis is to assess to what extent **contextual knowledge** can be exploited to improve the interactive behavior of robots in Human-Robot Interaction (HRI) tasks.

This thesis is motivated by the fact that in situated scenarios humans and robots make references to the environment; context is thus a valuable resource of knowledge that needs to be accounted to enable effective HRIs.

In order to define the boundaries of this research and frame the contributions within the literature, it is thus required to:

- identify a specific interaction modality to be leveraged in HRI,
- identify specific HRI problems where the interplay between the targeted interaction modality and context is expected to have an impact, and
- define different expressions of contextual knowledge with respect to both the individual HRI problem and the accounted interaction modality.

Since service robots are expected to efficiently interact with people, we focus on the humans' preferred vehicle of communication: Natural Language (NL). NL is expressive and efficient and enables natural communication. We call Spoken Human-Robot Interaction (SHRI), the specific sub-field of HRI where the interaction takes place through NL.

Hence, we focus on tasks where NL plays a key role. One of the most interesting paradigms in the range of HRI is represented by Symbiotic Autonomy, where humans and robots interact to help each other in achieving tasks. We thus aim at exploring all the directions of such a paradigm: the scenario in which the robot needs help, the situation where the human needs help and the case where the collaboration between the two actors is beneficial to accomplish a common task. In each of the

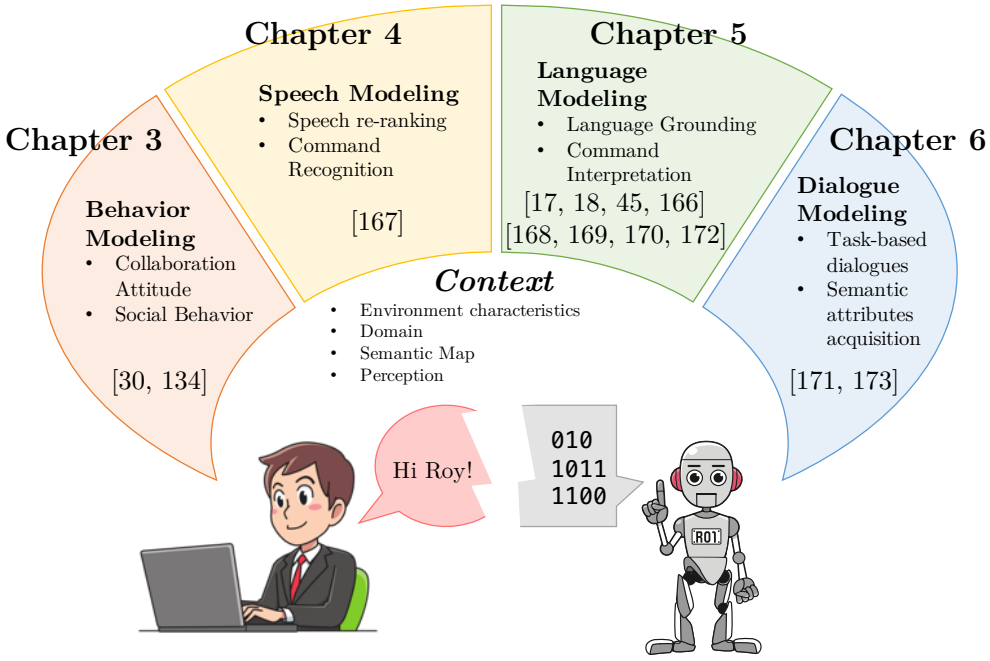


Figure 7.1. Interplay between context and Behavior, Language and Dialogue modeling in a Situated HRI

above scenarios, we identify issues and flaws that might be potentially overcome by a proper adoption of contextual knowledge. Hence, we first investigate to what extent the operational context introduces biases on the willingness of people in helping a robot to achieve its tasks. Then, we move to the problem of interpreting robotic commands in NL, improving the accuracy of the speech recognition process and solving language ambiguities through contextual knowledge about the domain and the specific Semantic Map. Finally, we explore how context may improve the dialogic interaction experience in a collaborative task, where the user instructs the robot to fill missing information of the Semantic Map.

Figure 7.1 recalls the role of the context in Situated SHRI, associating each contribution to the corresponding chapter and publications.

In the following, a brief recap of each contribution is provided.

7.1.1. Chapter 3: The Role of Context in Robot Behavior Modeling

This chapter focuses on evaluating the willingness of human in helping a robot, in the context of Symbiotic Autonomy. In this specific HRI scenario, robots ask humans for help, provided that robots exhibit proper social behaviors. To this end, we run two user studies to assess the contextual factors that may influence such Collaboration Attitude, gleaning from observable characteristics of the operating environment. In particular, *Proxemics* and *Gender* seem to have a strong influence on the users' attitude towards collaborating, where the Personal space of the user occupied by the robot seems to be the most comfortable one. On the other hand, our experimental data allow supporting the claim that females are more inclined to

collaborate. Instead, humans' *Height* needs to be further analyzed, in relation to the robot's size. Conversely, the *Operational Environment*, where the interaction takes place, and the *Activity*, that users are performing during the interaction, do not seem to impact on the Collaboration Attitude of humans. In conclusion, the overall study provides some insights on the contextual factors possibly influencing the Collaboration Attitude, that may help creating guidelines for designing robots' behavior.

Open Problems and Future Directions. This study focuses on a small subset of contextual factors influencing the willingness of users towards collaborating with a robotic platform. Nevertheless, a vast plethora of other factors are expected to influence the Collaboration Attitude of the users in the context of Symbiotic Autonomy and these factors are worth to be identified. For example, how the Collaboration Attitude varies between interactions within small groups of people and interactions with individuals, and how participants are influenced by different appearances or structures of the robot might be valuable subjects for the research on the design of robots' behaviors.

7.1.2. Chapter 4: The Role of Context in Speech Recognition

This chapter introduces a practical yet robust approach for improving the recognition capabilities of a generic Automatic Speech Recognition (ASR) when applied to a specific domain. In fact, starting from our need of interpreting robotic commands in the context of SHRI, we design a technique that takes into account domain-specific evidence to re-rank the transcriptions hypothesized by the ASR. In particular, a cost is assigned to each ASR transcription, that decreases along with the number of constraints satisfied by the sentence with respect to adopted grammar. The constraints are imposed by a grammar, designed to parse NL robotic commands. Experimental results show that, in spite of the simplicity of the proposed technique, it allows to significantly improve the performance of an open-domain ASR system, suggesting that the approach could be potentially applied to a real scenario.

Open Problems and Future Directions. This simple method could be jointly used with supervised learning methods, exploiting both pure linguistic features and evidence derived from the grammar, to learn more expressive re-ranking functions. Moreover, future works might consider re-ranking strategies over lists of interpretations, rather than hypotheses.

7.1.3. Chapter 5: The Role of Context in Language Modeling

This chapter presents a framework for the interpretation of robotic commands, in the context of SHRI. The Machine Learning (ML) processes underlying the system are designed to consider both linguistic observations of the sentence, as well as spatial and semantic information of the operating environment, extracted from a Semantic Map. This allows producing interpretations that are coherent with the environment and motivated by the operational context. Moreover, the logic forms corresponding to the meaning of commands are compliant with Frame Semantics, a well-established theory of linguistic meaning. The ML framework is implemented as

a cascade of Hidden Markov Support Vector Machines (SVM^{hmm}s) classifiers, each of which focused on a specific sub-problem of the whole interpretation process.

In order to adopt such a ML technique, we develop a corpus of annotated robotic commands, Human-Robot Interaction Corpus (HuRIC), to successfully train and test the language understanding framework. This corpus, originally composed by examples in English, has been improved by collecting also a subset of examples in Italian, and by pairing each command with its corresponding Semantic Map.

Experimental evaluations prove the effectiveness of the proposed solution, providing outstanding accuracy over two languages, i.e., English and Italian. Moreover, the outcomes prove the effect of contextual features extracted from the Semantic Map, which contributed, to a different extent, to the improvement of each sub-task.

Open Problems and Future Directions. Further effort is required to keep improving the process, starting from an extension of HuRIC with additional sentences, thus providing a wider coverage of the addressed linguistic phenomena, and including even more semantic features, in order to tighten even more the interpretation of the sentence to the operational environment. Future research will also focus on the extension of the proposed methodology, for example by considering more fine-grained spatial relations between entities in the environment or their physical characteristics, such as their color, in the grounding function. Furthermore, the same information may be encoded in a Deep Learning framework, to exploit the full power of such learning techniques.

In conclusion, the proposed solution can support further and more challenging research topics in the context of SHRI, such as interactive question answering or dialogue with robots.

7.1.4. Chapter 6: The Role of Context in Dialogue Modeling

This chapter addresses the problem of acquiring dialogue policies to support robot teaching tasks, for the minimization of the users' tutoring cost. The approach is based on a Dialogue Manager (DM) designed as a multi-objective Markov Decision Process (MDP), where the optimization problem is solved through Reinforcement Learning (RL) and the interaction is made dependent on **contextual visual information**. In fact, while a first MDP is used to predict the proper sentence the robot is supposed to utter, a second MDP modifies the trust towards visual context. Such information allows the robot to make predictions about the semantic properties to be acquired, relieving the user of further dialogic interactions. The visual classifier is implemented through an online incremental technique, based on Load-Balancing Self-Organizing Incremental Neural Network (LB-SOINN).

The experimental evaluation shows the effectiveness of the proposed solution, providing an empirical proof of the benefits introduced by the adaptive threshold MDP in minimizing the users' tutoring cost.

Open Problems and Future Directions. This work represents a starting point for a future line of research. First, the proposed online scheme, as well as its real-time processing, allows for a preliminary deployment of such system in a real scenario. This will enable a thorough evaluation of a real robot interacting with real users.

Second, the investigation of more accurate metrics to evaluate the reliability of the visual classifier (e.g., entropy, robustness, ...) could be beneficial for the policy acquisition. Finally, though the focus is on the category of an object, a larger set of semantic properties could also be taken into account (e.g., attributes such as *color*, *affordances*, ...), to fill all the missing information of a complete semantic map. To this end, different and more suited MDP design patterns can be explored and evaluated.

7.2. Thesis Statement and Final Remarks

This thesis explores the role of the context in SHRI, arguing that:

1. the interplay between context and interaction in NL is motivated by different reasons, generating different forms of contextual knowledge;

2. different forms of contextual knowledge can be exploited to improve the individual SHRI sub-tasks.

In fact, due to the context-aware nature of interactions, we proved that all the involved processes must take into account the operational context, in order to mimic humans cognitive processes. Such a claim is supported by the analysis of different expressions of context, applied to different SHRI tasks. In fact, context is expressed in different forms, that depend on the addressed task. Through extensive experimental evaluations it has been possible to prove the benefits brought by the proper use of context within the different tasks.

Bibliography

[1] Luigia Carlucci Aiello, Emanuele Bastianelli, Luca Iocchi, Daniele Nardi, Vittorio Perera, and Gabriele Randelli. Knowledgeable talking robots. In *Artificial General Intelligence - 6th International Conference, AGI 2013, Beijing, China, July 31 - August 3, 2013 Proceedings*, pages 182–191, 2013. doi: 10.1007/978-3-642-39521-5_21. (Cited on page 45.)

[2] Takako Aikawa, Chris Quirk, and Lee Schwartz. Learning prepositional attachment from sentence aligned bilingual corpora. Association for Machine Translation in the Americas, September 2003. (Cited on page 55.)

[3] Alyssa Alcorn, Helen Pain, Gnanathusharan Rajendran, Tim Smith, Oliver Lemon, Kaska Porayska-Pomsta, Mary Ellen Foster, Katerina Avramides, Christopher Frauenberger, and Sara Bernardini. Social communication between virtual characters and children with autism. In Gautam Biswas, Susan Bull, Judy Kay, and Antonija Mitrovic, editors, *Artificial Intelligence in Education*, pages 7–14, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-21869-9. (Cited on page 16.)

[4] M Alomari, P Duckworth, M Hawasly, DC Hogg, and AG Cohn. Natural language grounding and grammar induction for robotic manipulation commands, August 2017. (Cited on page 53.)

[5] Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden Markov support vector machines. In *Proceedings of the International Conference on Machine Learning*, 2003. (Cited on pages 60 and 145.)

[6] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The berkeley framenet project. In *Proceedings of ACL and COLING*, pages 86–90, 1998. (Cited on pages 57 and 71.)

[7] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria, August 2013. (Cited on page 81.)

[8] Mohit Bansal, Cynthia Matuszek, Jacob Andreas, Yoav Artzi, and Yonatan Bisk, editors. *Proceedings of the First Workshop on Language Grounding for Robotics*. Association for Computational Linguistics, Vancouver, Canada, August 2017. (Cited on page 52.)

[9] Roberto Basili and Fabio Massimo Zanzotto. Parsing engineering and empirical robustness. *Nat. Lang. Eng.*, 8(3):97–120, June 2002. ISSN 1351-3249. doi: 10.1017/S1351324902002875. (Cited on page 81.)

[10] Roberto Basili, Emanuele Bastianelli, Giuseppe Castellucci, Daniele Nardi, and Vittorio Perera. Kernel-based discriminative re-ranking for spoken command understanding in hri. In Matteo Baldoni, Cristina Baroglio, Guido Boella, and Roberto Micalizio, editors, *AI*IA 2013: Advances in Artificial Intelligence*, pages 169–180, Cham, 2013. Springer International Publishing. ISBN 978-3-319-03524-6. (Cited on page 81.)

[11] E. Bastianelli, D.D. Bloisi, R. Capobianco, F. Cossu, G. Gemignani, L. Iocchi, and D. Nardi. On-line semantic mapping. In *Advanced Robotics (ICAR), 2013 16th International Conference on*, pages 1–6, Nov 2013. doi: 10.1109/ICAR.2013.6766501. (Cited on page 17.)

[12] Emanuele Bastianelli, Giuseppe Castellucci, Danilo Croce, Roberto Basili, and Daniele Nardi. Effective and robust natural language understanding for human-robot interaction. In *Proceedings of ECAI 2014*. IOS Press, 2014. doi: 10.3233/978-1-61499-419-0-57. (Cited on page 57.)

[13] Emanuele Bastianelli, Giuseppe Castellucci, Danilo Croce, Roberto Basili, and Daniele Nardi. Huric: a human robot interaction corpus. In *Proceedings of LREC 2014*, Reykjavik, Iceland, may 2014. (Cited on pages 40, 71, and 75.)

[14] Emanuele Bastianelli, Danilo Croce, Roberto Basili, and Daniele Nardi. Using semantic models for robust natural language human robot interaction. In *AI* IA 2015, Advances in Artificial Intelligence*, pages 343–356. Springer International Publishing, 2015. (Cited on pages 59 and 60.)

[15] Emanuele Bastianelli, Danilo Croce, Roberto Basili, and Daniele Nardi. Using semantic maps for robust natural language interaction with robots. In *Sixteenth Annual Conference of the International Speech Communication Association*, pages 1393–1397. International Speech Communication Association, 2015. (Cited on pages xv, 41, 47, and 48.)

[16] Emanuele Bastianelli, Daniele Nardi, Luigia Carlucci Aiello, Fabrizio Giacomelli, and Nicolamaria Manes. Speaky for robots: the development of vocal interfaces for robotic applications. *Applied Intelligence*, 44(1):43–66, 2015. ISSN 1573-7497. (Cited on pages 42, 44, and 45.)

[17] Emanuele Bastianelli, Danilo Croce, Andrea Vanzo, Roberto Basili, and Daniele Nardi. A discriminative approach to grounded spoken language understanding in interactive robotics. In Subbarao Kambhampati, editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, IJCAI'16, pages 2747–2753, New York, New York, USA, July 2016. IJCAI/AAAI Press. (Cited on pages 60, 65, 69, 83, and 104.)

[18] Emanuele Bastianelli, Danilo Croce, Andrea Vanzo, Roberto Basili, and Daniele Nardi. Perceptually informed spoken language understanding for service robotics. In *Proceedings of the IJCAI2016 Workshop on Autonomous Mobile Service Robots*, New York City, US, 2016. (Cited on pages 75 and 104.)

[19] Emanuele Bastianelli, Giuseppe Castellucci, Danilo Croce, Roberto Basili, and Daniele Nardi. Structured learning for spoken language understanding in human-robot interaction. *The International Journal of Robotics Research*, 36 (5-7):660–683, 2017. doi: 10.1177/0278364917691112. (Cited on page 57.)

[20] Richard Bellman. A markovian decision process. *Journal of Mathematics and Mechanics*, 6(5):679–684, 1957. URL `http://www.jstor.org/stable/24900506`. (Cited on page 146.)

[21] Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1 edition, 1957. (Cited on page 148.)

[22] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March 2003. ISSN 1532-4435. (Cited on page 153.)

[23] Yevgeni Berzak, Andrei Barbu, Daniel Harari, Boris Katz, and Shimon Ullman. Do you see what I mean? visual resolution of linguistic ambiguities. *CoRR*, abs/1603.08079, 2016. (Cited on page 53.)

[24] Johan Bos and Tetsushi Oka. A spoken language interface with a mobile robot. *Artificial Life and Robotics*, 11(1):42–47, 2007. (Cited on page 17.)

[25] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787. (Cited on page 135.)

[26] Michael Brenner and Ivana Kruijff-Korbayová. A continual multiagent planning approach to situated dialogue. *Semantics and Pragmatics of Dialogue (LONDIAL)*, page 61, 2008. (Cited on page 18.)

[27] E. Brunskill, T. Kollar, and N. Roy. Topological mapping using spectral clustering and classification. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3491–3496, October 2007. (Cited on page 86.)

[28] P. Buschka and A. Saffiotti. A virtual sensor for room detection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 637–642, 2002. (Cited on pages 22 and 86.)

[29] Roberto Capobianco, Jacopo Serafin, Johann Dichtl, Giorgio Grisetti, Luca Iocchi, and Daniele Nardi. A proposal for semantic map representation and evaluation. In *Mobile Robots (ECMR), 2015 European Conference on*, pages 1–6. IEEE, 2015. (Cited on page 19.)

[30] Roberto Capobianco, Guglielmo Gemignani, Luca Iocchi, Daniele Nardi, Francesco Riccio, and Andrea Vanzo. Contexts for symbiotic autonomy: Semantic mapping, task teaching and social robotics. In Jeffrey O. Kephart, Stephanie Rosenthal, Manuela M. Veloso, and Alex Rudnicky, editors, *Symbiotic Cognitive Systems, Papers from the 2016 AAAI Workshop, Phoenix, Arizona, USA, February 13, 2016.*, volume WS-16-14 of *AAAI Workshops*, pages 733–736, Phoenix, Arizona, USA, February 2016. AAAI Press. (Cited on page 104.)

[31] S. Chandra, P. Alves-Oliveira, S. Lemaignan, P. Sequeira, A. Paiva, and P. Dillenbourg. Can a child feel responsible for another in the presence of a robot in a collaborative learning activity? In *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 167–172, Aug 2015. doi: 10.1109/ROMAN.2015.7333678. (Cited on page 16.)

[32] C. Chelba, Peng Xu, F. Pereira, and T. Richardson. Distributed acoustic modeling with back-off n-grams. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 4129–4132, Mar 2012. doi: 10.1109/ICASSP.2012.6288827. (Cited on page 46.)

[33] David L. Chen and Raymond J. Mooney. Learning to interpret natural language navigation instructions from observations. In *Proceedings of the 25th AAAI Conference on AI*, pages 859–865, 2011. (Cited on pages 17 and 57.)

[34] Gordon Christie, Ankit Laddha, Aishwarya Agrawal, Stanislaw Antol, Yash Goyal, Kevin Kochersberger, and Dhruv Batra. Resolving vision and language ambiguities together: Joint segmentation & prepositional attachment resolution in captioned scenes. *Computer Vision and Image Understanding*, 163:101 – 112, 2017. ISSN 1077-3142. Language in Vision. (Cited on page 53.)

[35] Kenneth Church and Ramesh Patil. Coping with syntactic ambiguity or how to put the block in the box on the table. *Computational Linguistics*, 8(3-4): 139–149, July 1982. ISSN 0891-2017. (Cited on page 55.)

[36] Herbert H. Clark and Susan E. Brennan. Grounding in communication. In Lauren Resnick, Levine B., M. John, Stephanie Teasley, and D., editors, *Perspectives on Socially Shared Cognition*, pages 13–1991. American Psychological Association, 1991. (Cited on pages 2 and 6.)

[37] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390177. (Cited on page 153.)

[38] Silvia Coradeschi and Alessandro Saffiotti. Symbiotic robotic systems: Humans, robots, and smart environments. *Intelligent Systems, IEEE*, 21(3):82–84, 2006. (Cited on pages 22 and 26.)

[39] K. Crammer and Y. Singer. On the algorithmic implementation of multi-class svms. *Journal of Machine Learning Research*, 2:265–292, 2001. (Cited on page 145.)

[40] Maartje M.A. de Graaf and Somaya Ben Allouch. Expectation setting and personality attribution in hri. In *Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction*, HRI '14, pages 144–145, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2658-2. doi: 10.1145/2559636.2559796. URL `http://doi.acm.org/10.1145/2559636.2559796`. (Cited on page 37.)

[41] Albert Diosi, Geoffrey R. Taylor, and Lindsay Kleeman. Interactive SLAM using laser and advanced sonar. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, ICRA 2005, April 18-22, 2005, Barcelona, Spain*, pages 1103–1108, 2005. (Cited on page 56.)

[42] Masrur Doostdar, Stefan Schiffer, and Gerhard Lakemeyer. *RoboCup 2008: Robot Soccer World Cup XII*, chapter A Robust Speech Recognition System for Service-Robotics Applications, pages 1–12. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-642-02921-9. doi: 10.1007/978-3-642-02921-9_1. (Cited on page 41.)

[43] T. M. Ellison. *CoNLL97 : Computational Natural Language Learning: Proceedings of the 1997 Meeting of the ACL Special Interest Group in Natural Language Learning*. Association for Computational Linguistics, 1997. (Cited on page 129.)

[44] Christoph Engel. Dictator games: a meta study. *Experimental Economics*, 14: 583–610, 2011. (Cited on pages 5, 7, and 36.)

[45] Daniele Evangelista, Wilson Umberto Villa, Marco Imperoli, Andrea Vanzo, Luca Iocchi, Daniele Nardi, and Alberto Pretto. Grounding natural language instructions in industrial robotics. In *Proceedings of the IROS 2017 Workshop "Human-Robot Interaction in Collaborative Manufacturing Environments (HRI-CME), Vancouver, Canada, September 24, 2017.*, Vancouver, Canada, 2017. (Cited on pages 75 and 104.)

[46] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008. (Cited on page 61.)

[47] Rui Fang, Changsong Liu, Lanbo She, and Joyce Y Chai. Towards situated dialogue: Revisiting referring expression generation. In *EMNLP*, pages 392–402, 2013. (Cited on page 18.)

[48] Rui Fang, Malcolm Doering, and Joyce Y Chai. Collaborative models for referring expression generation in situated dialogue. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1544–1550, 2014. (Cited on page 18.)

[49] Juan Fasola and Maja J Matarić. A socially assistive robot exercise coach for the elderly. *J. Hum.-Robot Interact.*, 2(2):3–32, June 2013. ISSN 2163-0364. doi: 10.5898/JHRI.2.2.Fasola. (Cited on page 16.)

[50] David Feil-Seifer and Maja Matarić. Ethical principles for socially assistive robotics. *IEEE Robotics and Automation Magazine*, 18(1):24–31, March 2011. doi: 10.1109/MRA.2010.940150. (Cited on page 16.)

[51] Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. Introducing and evaluating ukwac, a very large web-derived corpus of english. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google*, pages 47–54, 2008. (Cited on page 66.)

[52] Simone Filice, Giuseppe Castellucci, Danilo Croce, and Roberto Basili. Kelp: a kernel-based learning platform for natural language processing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL2015): System Demonstrations*, Beijing, China, 26-31 July 2015. (Cited on pages 61, 66, and 81.)

[53] Charles J. Fillmore. Frames and the semantics of understanding. *Quaderni di Semantica*, 6(2):222–254, 1985. (Cited on pages 8, 42, 57, and 71.)

[54] Julia Fink, Séverin Lemaignan, Pierre Dillenbourg, Philippe Rétornaz, Florian Vaussard, Alain Berthoud, Francesco Mondada, Florian Wille, and Karmen Franinović. Which robot behavior can motivate children to tidy up their toys?: Design and evaluation of "ranger". In *Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction*, HRI '14, pages 439–446, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2658-2. doi: 10.1145/2559636.2559659. (Cited on page 16.)

[55] Kerstin Fischer, Stephen Yang, Brian Mok, Rohan Maheshwari, David Sirkin, and Wendy Ju. Initiating interactions and negotiating approach: A robotic trash can in the field. In *AAAI Symposium on Turn-taking and Coordination in Human-Machine Interaction*, pages 10–16, 2015. (Cited on pages 16 and 26.)

[56] Mary Ellen Foster, Katerina Avramides, Sara Bernardini, Jingying Chen, Christopher Frauenberger, Oliver Lemon, and Kaska Porayska-Pomsta. Supporting children's social communication skills through interactive narratives with virtual characters. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 1111–1114, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-933-6. doi: 10.1145/1873951.1874163. URL `http://doi.acm.org/10.1145/1873951.1874163`. (Cited on page 16.)

[57] Mary Ellen Foster, Rachid Alami, Olli Gestranius, Oliver Lemon, Marketta Niemelä, Jean-Marc Odobez, and Amit Kumar Pandey. The MuMMER project: Engaging human-robot interaction in real-world public spaces. In *Proceedings of the Eighth International Conference on Social Robotics (ICSR 2016)*, 11 2016. doi: 10.1007/978-3-319-47437-3_74. (Cited on pages 1 and 16.)

[58] Shen Furao and Osamu Hasegawa. An incremental network for on-line unsupervised classification and topology learning. *Neural Netw.*, 19(1):90–106, January 2006. ISSN 0893-6080. doi: 10.1016/j.neunet.2005.04.006. URL `http://dx.doi.org/10.1016/j.neunet.2005.04.006`. (Cited on page 155.)

[59] Shen Furao, Tomotaka Ogura, and Osamu Hasegawa. An enhanced self-organizing incremental neural network for online unsupervised learning. *Neural Networks*, 20(8):893 – 903, 2007. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2007.07.008. (Cited on page 156.)

[60] Cipriano Galindo, Alessandro Saffiotti, Silvia Coradeschi, Pär Buschka, Juan-Antonio Fernandez-Madrigal, and Javier González. Multi-hierarchical semantic maps for mobile robotics. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 2278–2283. IEEE, 2005. (Cited on pages 22 and 86.)

[61] Qiaozi Gao, Malcolm Doering, Shaohua Yang, and Joyce Yue Chai. Physical causality of action verbs in grounded language understanding. In *ACL (1)*. The Association for Computer Linguistics, 2016. ISBN 978-1-945626-00-5. (Cited on page 53.)

[62] Konstantina Garoufi and Alexander Koller. Automated planning for situated natural language generation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1573–1582, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. (Cited on page 18.)

[63] Spandana Gella, Mirella Lapata, and Frank Keller. Unsupervised visual sense disambiguation for verbs using multimodal embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 182–192. Association for Computational Linguistics, 2016. doi: 10.18653/v1/N16-1022. (Cited on page 53.)

[64] Guglielmo Gemignani, Emanuele Bastianelli, and Daniele Nardi. Teaching robots parametrized executable plans through spoken interaction. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems*, 2015. (Cited on page 18.)

[65] Guglielmo Gemignani, Roberto Capobianco, and Daniele Nardi. Approaching qualitative spatial reasoning about distances and directions in robotics. In *14th Italian Conference on Artificial Intelligence*, 2015. (Cited on page 23.)

[66] Guglielmo Gemignani, Steven D. Klee, Manuela Veloso, and Daniele Nardi. On task recognition and generalization in long-term robot teaching. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, pages 1879–1880, Richland, SC, 2015. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-3413-6. (Cited on page 18.)

[67] Guglielmo Gemignani, Roberto Capobianco, Emanuele Bastianelli, Domenico Bloisi, Luca Iocchi, and Daniele Nardi. Living with robots: Interactive environmental knowledge acquisition. *Robotics and Autonomous Systems*, 78:1–16, 2016. doi: 10.1016/j.robot.2015.11.001. (Cited on pages 22, 23, 53, 56, and 87.)

[68] Nils Goerke and Sven Braun. Building semantic annotated maps by mobile robots. In *Proceedings of the Conference Towards Autonomous Robotic Systems*, pages 149–156, 2009. (Cited on page 86.)

[69] G. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics*, 2 (2), 1965. (Cited on page 153.)

[70] Barbara Gonsior, Dirk Wollherr, and Martin Buss. Towards a dialog strategy for handling miscommunication in human-robot dialog. In *RO-MAN*, pages 264–269. IEEE, 2010. (Cited on page 18.)

[71] Barbara Gonsior, Christian Landsiedel, Antonia Glaser, Dirk Wollherr, and Martin Buss. Dialog strategies for handling miscommunication in task-related HRI. In *RO-MAN*, pages 369–375. IEEE, 2011. (Cited on page 18.)

[72] Michael A. Goodrich and Alan C. Schultz. Human-robot interaction: A survey. *Found. Trends Hum.-Comput. Interact.*, 1(3):203–275, January 2007. ISSN 1551-3955. doi: 10.1561/1100000005. (Cited on page 15.)

[73] David Gouaillier, Vincent Hugel, Pierre Blazevic, Chris Kilner, Jérôme Monceaux, Pascal Lafourcade, Brice Marnier, Julien Serre, and Bruno Maisonnier. Mechatronic design of nao humanoid. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, ICRA'09, pages 2124–2129, Piscataway, NJ, USA, 2009. IEEE Press. ISBN 978-1-4244-2788-8. (Cited on page 17.)

[74] S. Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346, 1990. ISSN 0167-2789. (Cited on page 5.)

[75] Zellig Harris. Distributional structure. *Word*, 10(23):146–162, 1954. (Cited on page 151.)

[76] Stefan Heinrich and Stefan Wermter. Towards robust speech recognition for human-robot interaction. In *Proceedings of the IROS2011 Workshop on Cognitive Neuroscience Robotics (CNR)*, pages 23–28, Sep 2011. (Cited on page 41.)

[77] Joachim Hertzberg and Alessandro Saffiotti. Using semantic knowledge in robotics. *Robotics and Autonomous Systems*, 56(11):875–877, 2008. (Cited on pages 22 and 86.)

[78] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition. *Signal Processing Magazine*, 2012. (Cited on pages 39 and 99.)

[79] Deanna Hood, Séverin Lemaignan, and Pierre Dillenbourg. When children teach a robot to write: An autonomous teachable humanoid which uses simulated handwriting. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, HRI '15, pages 83–90, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-2883-8. doi: 10.1145/2696454.2696479. (Cited on page 16.)

[80] Andrew Hunt and Scott McGlashan. Speech recognition grammar specification. Technical report, World Wide Web Consortium, 2004. (Cited on page 45.)

[81] Luca Iocchi, M. T. Lázaro, Laurent Jeanpierre, Abdel-Illah Mouaddib, Esra Erdem, and Hichem Sahli. Coaches cooperative autonomous robots in complex and human populated environments. In Marco Gavanelli, Evelina Lamma, and Fabrizio Riguzzi, editors, *AI*IA 2015 Advances in Artificial Intelligence*, pages 465–477, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24309-2. (Cited on page 1.)

[82] Rebecca Jonson. Grammar-based context-specific statistical language modelling. In *Proceedings of the Workshop on Grammar-Based Approaches to Spoken Language Processing*, SLP '07, pages 25–32, Stroudsburg, PA, USA, 2007. (Cited on page 41.)

[83] P. Kankuekul, A. Kawewong, S. Tangruamsub, and O. Hasegawa. Online incremental attribute-based zero-shot learning. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3657–3664, June 2012. doi: 10.1109/CVPR.2012.6248112. (Cited on page 155.)

[84] F. Kaplan. Talking AIBO: First experimentation of verbal interactions with an autonomous four-legged robot. In *Proceedings of the CELE-Twente workshop on interacting agents*, 2000. (Cited on page 53.)

[85] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, and Eiichi Osawa. Robocup: The robot world cup initiative. In *Proceedings of the First International Conference on Autonomous Agents*, AGENTS '97, pages 340–347, New York, NY, USA, 1997. ACM. ISBN 0-89791-877-0. doi: 10.1145/267658.267738. URL `http://doi.acm.org/10.1145/267658.267738`. (Cited on pages 1 and 16.)

[86] Kheng Lee Koay, Dag Sverre Syrdal, Mohammadreza Ashgari-Oskoei, Michael L. Walters, and Kerstin Dautenhahn. Social roles and baseline proxemic preferences for a domestic service robot. *International Journal of Social Robotics*, 6:469–488, 2014. (Cited on pages 16, 27, and 31.)

[87] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78: 1464–1480, 1990. (Cited on page 89.)

[88] Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas Roy. Toward understanding natural language directions. In *Proceedings of the 5th ACM/IEEE International Conference on Human-robot Interaction*, HRI '10, pages 259–266, Piscataway, NJ, USA, 2010. IEEE Press. ISBN 978-1-4244-4893-7. (Cited on page 17.)

[89] Thomas Kollar, Vittorio Perera, Daniele Nardi, and Manuela M. Veloso. Learning environmental knowledge from task-based human-robot dialog. In *ICRA*, pages 4304–4309. IEEE, 2013. ISBN 978-1-4673-5641-1. (Cited on page 18.)

[90] Chen Kong, Dahua Lin, Mohit Bansal, Raquel Urtasun, and Sanja Fidler. What are you talking about? text-to-image coreference. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 3558–3565, Washington, DC, USA, 2014. IEEE Computer Society. ISBN 978-1-4799-5118-5. (Cited on page 53.)

[91] Evan Krause, Michael Zillich, Thomas Williams, and Matthias Scheutz. Learning to recognize novel objects in one shot through human-robot interactions in natural language dialogues. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, pages 2796–2802. AAAI Press, 2014. (Cited on page 155.)

[92] Brigitte Krenn and Christer Samuelsson. The linguist's guide to statistics - don't panic, 1997. (Cited on pages 141 and 143.)

[93] Jayant Krishnamurthy and Thomas Kollar. Jointly learning to parse and perceive: Connecting natural language to the physical world. *TACL*, 1:193–206, 2013. (Cited on page 54.)

[94] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*, pages 1106–1114, 2012. (Cited on page 89.)

[95] Geert-Jan M Kruijff, Hendrik Zender, Patric Jensfelt, and Henrik I Christensen. Clarification dialogues in human-augmented mapping. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 282–289. ACM, 2006. (Cited on pages 18, 23, and 87.)

[96] Geert-Jan M. Kruijff, H. Zender, P. Jensfelt, and Henrik I. Christensen. Situated dialogue and spatial organization: What, where... and why? *International Journal of Advanced Robotic Systems*, 4(2), 2007. (Cited on page 17.)

[97] G.M. Kruijff, H. Zender, P. Jensfelt, and H.I. Christensen. Situated dialogue and understanding spatial organization: Knowing what is where and what you can do there. In *Robot and Human Interactive Communication, 2006. ROMAN 2006. The 15th IEEE International Symposium on*, pages 328–333, Sept 2006. doi: 10.1109/ROMAN.2006.314438. (Cited on page 17.)

[98] Benjamin Kuipers and Yung-Tai Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems*, 8:47–63, 1991. (Cited on page 22.)

[99] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *2011 IEEE International Conference on Robotics and Automation*, pages 1817–1824, May 2011. doi: 10.1109/ICRA.2011.5980382. (Cited on pages 96 and 155.)

[100] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, March 2014. ISSN 0162-8828. doi: 10.1109/TPAMI.2013.140. (Cited on page 155.)

[101] T. Landauer and S. Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104(2):211–240, 1997. (Cited on page 153.)

[102] S. Lemaignan, A. Jacq, D. Hood, F. Garcia, A. Paiva, and P. Dillenbourg. Learning by teaching a robot: The case of handwriting. *IEEE Robotics Automation Magazine*, 23(2):56–66, June 2016. ISSN 1070-9932. doi: 10.1109/MRA.2016.2546700. (Cited on pages 1 and 16.)

[103] M. Levit, S. Chang, and B. Buntschuh. Garbage modeling with decoys for a sequential recognition scenario. In *Automatic Speech Recognition Understanding, 2009. ASRU 2009. IEEE Workshop on*, pages 468–473, Nov 2009. doi: 10.1109/ASRU.2009.5372919. (Cited on page 41.)

[104] Qiguang Lin, David Lubensky, Michael Picheny, and P. Srinivasa Rao. Key-phrase spotting using an integrated language model of n-grams and finite-state grammar. In George Kokkinakis, Nikos Fakotakis, and Evangelos Dermatas, editors, *EUROSPEECH*. ISCA, 1997. (Cited on page 41.)

[105] Diego Linares, José-Miguel Benedí, and Joan-Andreu Sánchez. A hybrid language model based on a combination of n-grams and stochastic context-free grammars. *ACM Transactions on Asian Language Information Processing*, 3 (2):113–127, Jun 2004. ISSN 1530-0226. doi: 10.1145/1034780.1034783. (Cited on page 41.)

[106] Peter Lindes, Aaron Mininger, James R Kirk, and John E Laird. Grounding language for interactive task learning. In *Proceedings of the First Workshop on Language Grounding for Robotics*, pages 1–9, 2017. (Cited on page 52.)

[107] Bing Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Data-Centric Systems and Applications. Springer, 2007. ISBN 978-3-540-37881-5. (Cited on page 143.)

[108] Changsong Liu, Jacob Walker, and Joyce Y. Chai. Ambiguities in spatial language understanding in situated human robot dialogue. In *AAAI Fall Symposium: Dialog with Robots*, volume FS-10-05 of *AAAI Technical Report*. AAAI, 2010. (Cited on page 18.)

[109] James MacGlashan. Burlap, 2015. URL `http://burlap.cs.brown.edu/`. (Cited on page 97.)

[110] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, Baltimore, Maryland, USA, June 22-27 2014. (Cited on page 81.)

[111] Matthew Marge and Alexander I. Rudnicky. Towards overcoming miscommunication in situated dialogue by asking questions. In *AAAI Fall Symposium: Building Representations of Common Ground with Intelligent Agents*, volume FS-11-02 of *AAAI Technical Report*. AAAI, 2011. (Cited on page 18.)

[112] Matthew Marge and Alexander I. Rudnicky. Miscommunication recovery in physically situated dialogue. In *SIGDIAL Conference*, pages 22–31. The Association for Computer Linguistics, 2015. (Cited on page 18.)

[113] Cynthia Matuszek, Nicholas FitzGerald, Luke S. Zettlemoyer, Liefeng Bo, and Dieter Fox. A joint model of language and perception for grounded attribute learning. In *ICML*. icml.cc / Omnipress, 2012. (Cited on page 54.)

[114] Cynthia Matuszek, Evan Herbst, Luke S. Zettlemoyer, and Dieter Fox. Learning to parse natural language commands to a robot control system. In Jaydev P. Desai, Gregory Dudek, Oussama Khatib, and Vijay Kumar, editors, *ISER*, volume 88 of *Springer Tracts in Advanced Robotics*, pages 403–415. Springer, 2012. (Cited on pages 17 and 57.)

[115] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013. (Cited on pages 53, 66, and 153.)

[116] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, nov 1995. ISSN 0001-0782. doi: 10.1145/219717.219748. (Cited on pages 75 and 150.)

[117] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997. ISBN 0070428077, 9780070428072. (Cited on page 129.)

[118] N. Mitsunaga, C. Smith, T. Kanda, H. Ishiguro, and N. Hagita. Adapting robot behavior for human–robot interaction. *IEEE Transactions on Robotics*, 24:911–916, 2008. (Cited on pages 5, 7, 16, 27, and 32.)

[119] F. Morbini, K. Audhkhasi, R. Artstein, M. Van Segbroeck, K. Sagae, P. Georgiou, D.R. Traum, and S. Narayanan. A reranking approach for recognition and classification of speech input in conversational dialogue systems. In *Spoken Language Technology Workshop (SLT), 2012 IEEE*, pages 49–54, Dec 2012. doi: 10.1109/SLT.2012.6424196. (Cited on page 40.)

[120] O. M. Mozos and W. Burgard. Supervised learning of topological maps using semantic information extracted from range data. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2772–2777, October 2006. (Cited on page 86.)

[121] Oscar Martinez Mozos, Hitoshi Mizutani, Ryo Kurazume, and Tsutomu Hasegawa. Categorization of indoor places using the kinect sensor. *Sensors*, 12(5):6695–6711, May 2012. (Cited on pages 22 and 86.)

[122] Jonathan Mumm and Bilge Mutlu. Human-robot proxemics: Physical and psychological distancing in human-robot interaction. In *Proceedings of the 6th International Conference on Human-robot Interaction*, HRI '11, pages 331–338. ACM, 2011. (Cited on pages 7, 16, 27, 32, and 36.)

[123] Carlos Nieto-Granda, John G Rogers III, Alexander JB Trevor, and Henrik Christensen. Semantic map partitioning in indoor environments using regional analysis. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1451–1456. IEEE, 2010. (Cited on pages 23 and 87.)

[124] Stefanos Nikolaidis, Ramya Ramakrishnan, Keren Gu, and Julie Shah. Efficient model learning from joint-action demonstrations for human-robot collaborative tasks. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, HRI '15, pages 189–196. ACM, 2015. (Cited on pages 16 and 26.)

[125] A. B. Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, volume 12, pages 615–622, New York, NY, USA, 1962. Polytechnic Institute of Brooklyn. (Cited on page 134.)

[126] Andreas Nüchter and Joachim Hertzberg. Towards semantic maps for mobile robots. *Robot. Auton. Syst.*, 56(11):915–926, 2008. (Cited on pages 4, 52, 56, and 86.)

[127] Dejan Pangercic, Moritz Tenorth, Benjamin Pitzer, and Michael Beetz. Semantic object maps for robotic housework - representation, acquisition and use. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vilamoura, Portugal, October, 7–12 2012. (Cited on page 22.)

[128] Jose L. Part and Oliver Lemon. Incremental online learning of objects for robots operating in real environments. In *Joint IEEE International Conference on Development and Learning and on Epigenetic Robotics (ICDL-EPIROB)*, Lisbon, Portugal, September 2017. (Cited on page 89.)

[129] David Paulk, Vangelis Metsis, Christopher McMurrough, and Fillia Makedon. A supervised learning approach for fast object recognition from rgb-d data. In *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '14, pages 5:1–5:8, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2746-6. doi: 10.1145/2674396.2674432. (Cited on page 155.)

[130] Andrzej Pronobis and Patric Jensfelt. Large-scale semantic mapping and reasoning with heterogeneous modalities. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3515–3522. IEEE, 2012. (Cited on pages 18, 23, and 87.)

[131] Leixian Qiao, Xue Li, and Shuqiang Jiang. Rgb-d object recognition from handheld object teaching. In *Proceedings of the International Conference on Internet Multimedia Computing and Service*, ICIMCS'16, pages 31–34, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4850-8. doi: 10.1145/3007669.3007713. (Cited on page 155.)

[132] Shaolin Qu and Joyce Yue Chai. Context-based word acquisition for situated dialogue in a virtual world. *Journal of Artificial Intelligence Research*, pages 247–277, 2010. (Cited on page 18.)

[133] Gabriele Randelli, Taigo Maria Bonanni, Luca Iocchi, and Daniele Nardi. Knowledge acquisition through human–robot multimodal interaction. *Intelligent Service Robotics*, 6(1):19–31, 2013. (Cited on page 23.)

[134] Francesco Riccio, Andrea Vanzo, Valeria Mirabella, Tiziana Catarci, and Daniele Nardi. *Enabling Symbiotic Autonomy in Short-Term Interactions: A User Study*, pages 796–807. Springer International Publishing, 2016. ISBN 978-3-319-47437-3. doi: 10.1007/978-3-319-47437-3_78. (Cited on pages 7, 25, and 104.)

[135] Markus Rickert, Mary Ellen Foster, Manuel Giuliani, Tomas By, Giorgio Panin, and Alois Knoll. Integrating language, vision and action for human robot dialog systems. In Constantine Stephanidis, editor, *Universal Access in Human-Computer Interaction. Ambient Interaction*, pages 987–995, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-73281-5. (Cited on page 18.)

[136] Laurel D. Riek. Wizard of oz studies in hri: A systematic review and new reporting guidelines. *J. Hum.-Robot Interact.*, 1(1):119–136, july 2012. ISSN 2163-0364. doi: 10.5898/JHRI.1.1.Riek. (Cited on page 30.)

[137] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958. (Cited on page 133.)

[138] Stephanie Rosenthal and Manuela Veloso. Mobile robot planning to seek help with spatially-situated tasks. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, pages 2067–2073. AAAI Press, 2012. (Cited on page 27.)

[139] Stephanie Rosenthal, Joydeep Biswas, and Manuela Veloso. An effective personal mobile robot agent through symbiotic human-robot interaction. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 915–922. International Foundation for Autonomous Agents and Multiagent Systems, 2010. (Cited on pages 22, 26, and 27.)

[140] Robert J Ross and John Bateman. Agency & information state in situated dialogues: Analysis & computational modelling. *DiaHolmia*, page 113, 2009. (Cited on page 18.)

[141] G. A. Rummery and M. Niranjan. On-line Q-learning using connectionist systems. Technical Report TR 166, Cambridge University Engineering Department, Cambridge, England, 1994. (Cited on page 149.)

[142] Paul E. Rybski, Kevin Yoon, Jeremy Stolarz, and Manuela M Veloso. Interactive robot task training through dialog and demonstration. In *Proceedings of the ACM/IEEE International Conference on Human-robot Interaction*, HRI '07, pages 49–56, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-617-2. doi: 10.1145/1228716.1228724. (Cited on page 18.)

[143] Magnus Sahlgren. *The Word-Space Model*. PhD thesis, Stockholm University, 2006. (Cited on pages 53, 151, and 152.)

[144] Giampiero Salvi and Stéphane Dupont, editors. *Proceedings GLU 2017 International Workshop on Grounding Language Understanding*. Stockholm, Sweden, August 2017. doi: 10.21437/GLU.2017. (Cited on page 52.)

[145] Matthias Scheutz, Rehj Cantrell, and Paul W. Schermerhorn. Toward human-like task-based dialogue processing for human robot interaction. *AI Magazine*, 32(4):77–84, 2011. (Cited on page 18.)

[146] Sven Schneider, Frederik Hegger, Aamir Ahmad, Iman Awaad, Francesco Amigoni, Jakob Berghofer, Rainer Bischoff, Andrea Bonarini, Rhama Dwiputra, Giulio Fontana, Luca Iocchi, Gerhard Kraetzschmar, Pedro Lima, Matteo Matteucci, Daniele Nardi, and Viola Schiaffonati. The rockin@home challenge. In *Proceedings of the 41st International Symposium on Robotics (ISR/Robotik 2014)*, pages 1–7, Munich, Germany, June 2-3 2014. (Cited on pages 79 and 81.)

[147] Hinrich Schütze. Word space. In *Advances in Neural Information Processing Systems 5*, 1993. (Cited on page 151.)

[148] Hinrich Schütze and Jan Pedersen. Information retrieval based on word senses. In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, USA, 1995. (Cited on page 151.)

[149] M. Schwarz, H. Schulz, and S. Behnke. Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1329–1335, May 2015. doi: 10.1109/ICRA.2015.7139363. (Cited on page 155.)

[150] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521813972. (Cited on page 138.)

[151] Danijel Skočaj, Matej Kristan, Alen Vrečko, Marko Mahnič, Miroslav Janíček, Geert-Jan M Kruijff, Marc Hanheide, Nick Hawes, Thomas Keller, Michael Zillich, et al. A system for interactive learning in dialogue with a tutor. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3387–3394. IEEE, 2011. (Cited on pages 22 and 155.)

[152] Danijel Skočaj, Matej Kristan, and Aleš Leonardis. Formalization of different learning strategies in a continuous learning framework. In *Proceedings of the Ninth International Conference on Epigenetic Robotics; Modeling Cognitive Development in Robotic Systems*, pages 153–160. Lund University Cognitive Studies, 2009. (Cited on pages 90 and 95.)

[153] Richard Socher, Brody Huval, Bharath Bhat, Christopher D. Manning, and Andrew Y. Ng. Convolutional-recursive deep learning for 3d object classification. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 656–664, USA, 2012. Curran Associates Inc. (Cited on page 155.)

[154] Hyang-gi Song, Michael Restivo, Arnout van de Rijt, Lori L. Scarlatos, David Tonjes, and Alex Orlov. The hidden gender effect in online collaboration: An experimental study of team performance under anonymity. *Computers in Human Behavior*, 50:274–282, 2015. (Cited on pages 7 and 36.)

[155] Laura Stoia, Darla Magdalene Shockley, Donna K. Byron, and Eric Fosler-Lussier. Noun phrase generation for situated dialogs. In *Proceedings of the Fourth International Natural Language Generation Conference*, INLG '06, pages 81–88, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. ISBN 1-932432-72-8. (Cited on page 18.)

[156] Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca A. Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *CoRR*, cs.CL/0006023, 2000. (Cited on page 89.)

[157] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998. ISBN 0262193981. (Cited on pages 97, 148, and 149.)

[158] D. S. Syrdal, K. Lee Koay, M. L. Walters, and K. Dautenhahn. A personalized robot companion? - the role of individual differences on spatial preferences in hri scenarios. In *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on*, pages 1143–1148, Aug 2007. (Cited on pages 7, 27, and 36.)

[159] L. Takayama and C. Pantofaru. Influences on proxemic behaviors in human-robot interaction. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 5495–5502, Oct 2009. (Cited on pages 16, 27, 32, and 36.)

[160] M. Tanenhaus, M. Spivey-Knowlton, K. Eberhard, and J. Sedivy. Integration of visual and linguistic information during spoken language comprehension. *Science*, 268:1632–1634, 1995. (Cited on pages 3 and 17.)

[161] S. Tellex, T. Kollar, S. Dickerson, M.R. Walter, A.G. Banerjee, S. Teller, and N. Roy. Approaching the symbol grounding problem with probabilistic graphical models. *AI Magazine*, 32(4), 2011. (Cited on pages 17 and 52.)

[162] Jesse Thomason, Shiqi Zhang, Raymond Mooney, and Peter Stone. Learning to interpret natural language commands through human-robot dialog. In *Proceedings of the 2015 International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1923–1929, Buenos Aires, Argentina, July 2015. (Cited on pages 18 and 53.)

[163] Jesse Thomason, Aishwarya Padmakumar, Jivko Sinapov, Justin Hart, Peter Stone, and Raymond J. Mooney. Opportunistic active learning for grounding natural language descriptions. In Sergey Levine, Vincent Vanhoucke, and Ken Goldberg, editors, *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pages 67–76. PMLR, 13–15 Nov 2017. (Cited on page 87.)

[164] Elena Torta, Raymond H. Cuijpers, and James F. Juola. A model of the user's proximity for bayesian inference. In *Proceedings of the 6th International Conference on Human-robot Interaction*, HRI '11, pages 273–274, New York, NY, USA, 2011. ACM. (Cited on page 31.)

[165] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, pages 104–, New York, NY, USA, 2004. ACM. ISBN 1-58113-838-5. (Cited on page 143.)

[166] Andrea Vanzo, Danilo Croce, Roberto Basili, and Daniele Nardi. Context-aware spoken language understanding for human robot interaction. In Pierpaolo Basile, Anna Corazza, Franco Cutugno, Simonetta Montemagni, Malvina Nissim, Viviana Patti, Giovanni Semeraro, and Rachele Sprugnoli, editors, *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016), Napoli, Italy, December 5-7, 2016.*, volume 1749 of *CEUR Workshop Proceedings*, pages

308–313, Napoli, Italy, December 2016. CEUR-WS.org. (Cited on pages 60, 83, and 104.)

[167] Andrea Vanzo, Danilo Croce, Emanuele Bastianelli, Roberto Basili, and Daniele Nardi. Robust spoken language understanding for house service robots. *Polibits*, 54:11–16, July 2016. doi: 10.17562/PB-54-2. (Cited on pages 40 and 104.)

[168] Andrea Vanzo, Danilo Croce, Giuseppe Castellucci, Roberto Basili, and Daniele Nardi. Spoken language understanding for service robotics in italian. In Giovanni Adorni, Stefano Cagnoni, Marco Gori, and Marco Maratea, editors, *AI*IA 2016: Advances in Artificial Intelligence - XVth International Conference of the Italian Association for Artificial Intelligence, Genova, Italy, November 29 - December 1, 2016, Proceedings*, volume 10037, pages 477–489, Genova, Italy, November 2016. Springer. ISBN 978-3-319-49130-1. doi: 10.1007/978-3-319-49130-1_35. (Cited on pages 66, 71, and 104.)

[169] Andrea Vanzo, Danilo Croce, Roberto Basili, and Daniele Nardi. Lu4r: adaptive spoken language understanding for robots. *Italian Journal of Computational Linguistics*, 3(1):59–76, June 2017. (Cited on pages 71, 75, and 104.)

[170] Andrea Vanzo, Danilo Croce, Roberto Basili, and Daniele Nardi. Structured learning for context-aware spoken language understanding of robotic commands. In Mohit Bansal, Cynthia Matuszek, Jacob Andreas, Yoav Artzi, and Yonatan Bisk, editors, *Proceedings of the First Workshop on Language Grounding for Robotics, Vancouver, Canada, August 3, 2017.*, pages 25–34, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-2804. (Cited on pages 60, 83, and 104.)

[171] Andrea Vanzo, Danilo Croce, Emanuele Bastianelli, Guglielmo Gemignani, Roberto Basili, and Daniele Nardi. Dialogue with robots to support symbiotic autonomy. In Kristiina Jokinen and Graham Wilcock, editors, *Dialogues with Social Robots - Enablements, Analyses, and Evaluation, Seventh International Workshop on Spoken Dialogue Systems, IWSDS 2016, Saariselkä, Finland, January 13-16, 2016*, volume 427 of *Lecture Notes in Electrical Engineering*, pages 331–342, Singapore, January 2017. Springer Singapore. ISBN 978-981-10-2585-3. doi: 10.1007/978-981-10-2585-3_27. (Cited on pages 87 and 104.)

[172] Andrea Vanzo, Luca Iocchi, Daniele Nardi, Raphael Memmesheimer, Dietrich Paulus, Iryna Ivanovska, and Gerhard K. Kraetzschmar. Benchmarking speech understanding in service robotics. In *4th International Workshop on Artificial Intelligence and Robotics (AIxIA)*, volume 2054 of *CEUR Workshop Proceedings*, pages 34–40. CEUR-WS.org, 2017. (Cited on pages 75 and 104.)

[173] Andrea Vanzo, Jose L. Part, Yanchao Yu, Daniele Nardi, and Oliver Lemon. Incrementally learning semantic attributes through dialogue interaction. In *Proceedings of the 17th Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, page To appear. International Foundation for Autonomous Agents and Multiagent Systems, 2018. (Cited on pages 9, 23, 85, and 104.)

[174] V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971. doi: 10.1137/1116025. (Cited on page 132.)

[175] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998. (Cited on page 135.)

[176] Marilyn A. Walker, Diane J. Litman, Candace A. Kamm, and Alicia Abella. Paradise: A framework for evaluating spoken dialogue agents. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, ACL '98, pages 271–280, Stroudsburg, PA, USA, 1997. Association for Computational Linguistics. (Cited on pages 93 and 95.)

[177] Michael L Walters, Kerstin Dautenhahn, René Te Boekhorst, Kheng Lee Koay, Dag Sverre Syrdal, and Chrystopher L Nehaniv. An empirical framework for human-robot proxemics. In *Proceedings of the Symposium on New Frontiers in Human-Robot Interaction*, pages 144–149, Edinburgh, Scottland, 2009. (Cited on pages 16, 27, and 36.)

[178] Thomas Wasow, Amy Perfors, and David Beaver. The puzzle of ambiguity. *Morphology and the web of grammar: Essays in memory of Steven G. Lapointe*, pages 265–282, 2005. (Cited on pages 5 and 55.)

[179] T. Winograd. Procedures as a representation for data in a computer program for understanding natural language. *Cognitive Psychology*, 3(1):1–191, 1972. (Cited on page 17.)

[180] T. Wisspeintner, T. van der Zant, L. Iocchi, and S. Schiffer. RoboCup@Home: Scientific competition and benchmarking for domestic service robots. *Interaction Studies*, 10(3):392–426, 2009. ISSN 1572-0373. (Cited on page 46.)

[181] J. Wu, H. I. Christensen, and J. M. Rehg. Visual place categorization: Problem, dataset, and algorithm. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4763–4770, October 2009. (Cited on pages 22 and 86.)

[182] Xiangyang Xu, Yuncheng Li, Gangshan Wu, and Jiebo Luo. Multi-modal deep feature learning for rgb-d object detection. *Pattern Recogn.*, 72(C):300–313, December 2017. ISSN 0031-3203. doi: 10.1016/j.patcog.2017.07.026. (Cited on page 155.)

[183] Shaohua Yang, Qiaozi Gao, Changsong Liu, Caiming Xiong, Song-Chun Zhu, and Joyce Y. Chai. Grounded semantic role labeling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 149–159. Association for Computational Linguistics, 2016. (Cited on page 53.)

[184] Yanchao Yu, Arash Eshghi, and Oliver Lemon. Training an adaptive dialogue policy for interactive learning of visually grounded word meanings. In *Proceedings of the SIGDIAL 2016 Conference*, pages 339–349. Association for Computational Linguistics, 9 2016. (Cited on page 90.)

[185] Yanchao Yu, Arash Eshghi, and Oliver Lemon. Learning how to learn: An adaptive dialogue agent for incrementally learning visually grounded word meanings. In *Proceedings of the First Workshop on Language Grounding*

for Robotics, pages 10–19, Vancouver, Canada, August 2017. Association for Computational Linguistics. (Cited on page 54.)

[186] Yanchao Yu, Arash Eshghi, Gregory Mills, and Oliver Lemon. The burchak corpus: a challenge data set for interactive learning of visually grounded word meanings. In *Proceedings of the Sixth Workshop on Vision and Language*, pages 1–10. Association for Computational Linguistics, 2017. (Cited on page 96.)

[187] Hendrik Zender, O Martínez Mozos, Patric Jensfelt, G-JM Kruijff, and Wolfram Burgard. Conceptual spatial representations for indoor mobile robots. *Robotics and Autonomous Systems*, 56(6):493–502, 2008. (Cited on pages 18, 22, and 87.)

[188] H. Zhang, X. Xiao, and O. Hasegawa. A load-balancing self-organizing incremental neural network. *IEEE Transactions on Neural Networks and Learning Systems*, 25(6):1096–1105, June 2014. ISSN 2162-237X. doi: 10.1109/TNNLS.2013.2287884. (Cited on pages 89, 90, 155, and 156.)

[189] Xiang Zuo, Naoto Iwahashi, Kotaro Funakoshi, Mikio Nakano, Ryo Taguchi, Shigeki Matsuda, Komei Sugiura, and Natsuki Oka. Detecting robot-directed speech by situated understanding in physical interaction. *Transactions of the Japanese Society for Artificial Intelligence*, 25(6):670–682, 2010. doi: 10.1527/tjsai.25.670. (Cited on page 17.)

Appendix A

Technical Preliminaries

A.1. Machine Learning for Spoken Human-Robot Interaction

One of the most valuable formal definition of what Machine Learning (ML) is, has been provided by Tom Mitchell [117]:

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

omputational Natural Language Learning [43] application, such formulation allows to define learning systems that can be applied to Natural Language Processing (NLP) problems. In detail:

- T represents a linguistic task, usually an interpretation process, such as in semantic annotation, document classification or opinion mining tasks. The objective is the acquisition from data of a classification function $y = f(x)$ able to map a text x into its corresponding class y.
- P represents the performance of f, that allow to evaluate the quality of the resulting computation. It depends on the task objectives and the learning system requirements.
- E represents data, that are used as available evidence about the target task. The idea is that a learning system exploits such information in order to acquire competences to resolve the target problem and the more information are observed, the highest improvements of the performance P to solve the task T are expected.

ML is thus a branch of Artificial Intelligence (AI) that refers to the design and study of systems that can learn from data, i.e., make optimal use of E to optimize P in solving T.

To this end, the core of any ML problem deals with two main sub-problems, namely *representation* and *generalization*. Representation is a crucial part of a ML system. It refers to the way data instances and their properties are represented.

Hence, it establishes the boundaries of what can be observed as part of E, what are the features that mostly fit the phenomena of the addressed task T and what can be made available to learn a function, i.e., perform T. Conversely, generalization is the property that a system should guarantee on unseen data instances; the conditions under which this can be done are a key object of study in the subfield of computational learning theory.

This Section explores and sketch some of the ML techniques supporting specific aspects of Spoken Human-Robot Interaction (SHRI). In particular, the focus will be on the approaches that have been used to draw this thesis up.

A.1.1. An Introduction to Supervised Learning

The idea underneath supervised learning is that each training example is provided with its corresponding label. In most cases, labels are manually generated by human annotators which estimate a label for each example, by exploiting their knowledge and reasoning skills. Hence, this family of learning algorithm is said to be supervised as the process of an algorithm learning from the training dataset can be seen as a teaching process: given the correct answer, the algorithm iteratively produces predictions on the training data that are, in turn, corrected by the teacher. The learning process stops whenever the algorithm achieves an acceptable level of performance.

In this respect, these methods require examples of pairs (x, y), where $x \in \mathcal{X}$ represents the observation, i.e., the set of aspects that are useful to characterize a concept, whereas $y \in \mathcal{Y}$ represents the correct label to which x can be mapped into. These pairs are exploited by the learning procedure to generate an approximation of the concept to be learned, by suggesting which are the features that better represent the concept itself.

The following subsections discuss and clarify supervised methods exploited in this thesis.

Support Vector Machines.

A discriminative algorithm learns models able to discriminate novel examples starting from their available observations $x \in \mathcal{X}$ and the corresponding labels $y \in \mathcal{Y}$. The algorithm finds an approximation $h(x) = \hat{y}$ of the ideal function $f(x) : \mathcal{T} \to \mathcal{Y}$ that, for each novel example x, is able to predict its correct label y, i.e., $h(x) \approx f(x)$. Hence, the learning algorithm acquires a set of possible mappings $x \mapsto f(x, \theta)$, where the functions $f(x, \theta)$ themselves are characterized by the parameters vector θ. Such automatic tagging process is assumed to be deterministic: it will always output the same prediction $f(x, \theta)$ for a given pair $\langle x, \theta \rangle$, generating what is called a trained machine. For example, according to a binary classification schema, for all x and θ the classifier produces the function $f(x, \theta) \in \{-1, 1\}$, being $f(x, \theta) = 1$ whenever x belongs to the target class y, while $f(x, \theta) = -1$ the opposite.

PAC-learning. Usually, training examples are randomly extracted and they often are not a representative sample of the classification function f. A function f is said to be a Probably Approximately Correct (PAC) learnable function and the underneath algorithm A is a Probably Approximately Correct learning algorithm, if

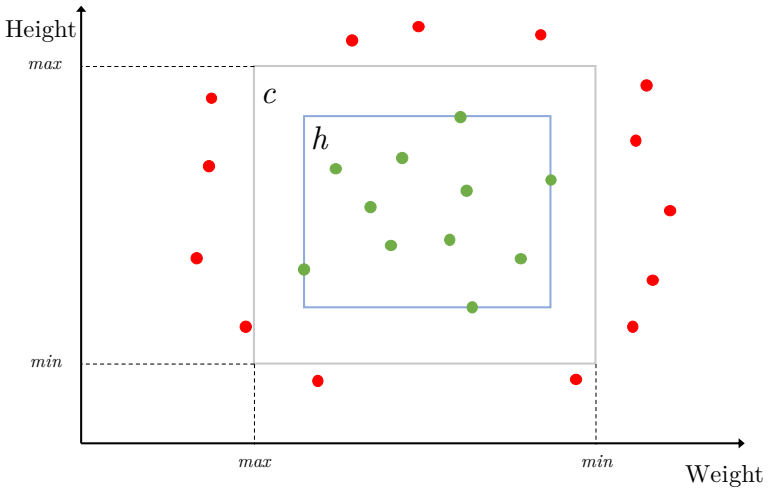


Figure A.1. PAC learning: example of the concept *"Average Build"*

A produces a hypothesis function $h \in \mathcal{H}$ that is "good approximation" of f, where $\mathcal{H}$ is the set of all possible hypothesis.

A formal definition of PAC-learning is provided in the following.

Definition A.1 (PAC-learning)**.** *Let $f : \mathcal{X} \to \mathcal{Y}, f \in \mathcal{F}$ be the function to learn. Let D be the probability distribution over $\mathcal{X}$. Let $h \in \mathcal{H}$ be the hypothesis function. Let*

$$Err(h, f) \equiv P_{x \in D}(h(x) \neq f(x)) \tag{A.1}$$

be the approximation error of h w.r.t. f.

Then, the class of functions $\mathcal{F}$ is PAC-learnable iff there exists an algorithm A such that, $\forall f \in \mathcal{F}, \forall D, \forall \epsilon > 0$ and $\forall \delta : 0 < \delta < 1$, A outputs h such that:

$$P(Err(h, f) > \epsilon) < \delta \tag{A.2}$$

Operationally, a class of functions $\mathcal{F}$ is *PAC-learnable* if, given a sizable training set, there exists an algorithm A that outputs a hypothesis h such that there is a probability less than δ that its error is greater than ϵ.

A practical example is provided in Figure A.1. The concept (or function) to learn is "Average Build", related to the Height and Weight characteristics (or *features*) of the subjects. The green points are the positive examples for the average build class, while the red ones are the negative examples. In order to select only the positive examples, the easiest way is to define h as the smallest rectangle able to contain them. Hence, the blue rectangle represents our hypothesis h, while the grey one is the concept to acquire c. The area between the rectangles is the hypothesis error. Therefore, we might say that h is an approximation of the concept c with a certain level of error.

In order to improve the hypothesis approximation, an oversized training set is required, so that some examples will be located into the error area. Hence, the training set size is strictly related to the approximation goodness: usually, the bigger is the training set, the smaller will be the error.

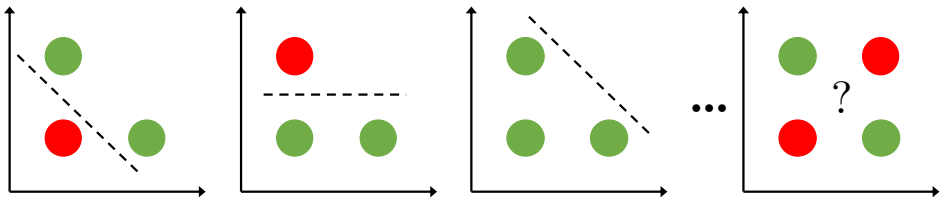


Figure A.2. Points in $\mathbb{R}^2$ shattered by separating hyperplanes

VC-dimension. However, some considerations about PAC learning are required. First, as already said, the number n of training examples considerably affects the error rate of the hypothesis function h. Second, the ability to learn a specific concept/function is highly influenced by the learning algorithm itself. Referring to the example in Figure A.1, a class of linear functions $\mathcal{F}$, instead of rectangular ones, would not be capable of correctly separating positive and negative examples. This essential property is formally defined through the *VC-dimension* [174].

Before introducing the *VC-dimension*, the definition of shattered set is provided below.

Definition A.2 (Shattered sets)**.** *Let $\mathcal{F}$ be a class of binary functions (that is, $\forall f \in \mathcal{F}, f : \mathcal{X} \to \{0,1\}$). Then, $\mathcal{F}$ is said to shatter $\mathcal{S} \subseteq \mathcal{X}$, if $\forall \mathcal{S}' \subseteq \mathcal{S}$, $\exists f \in \mathcal{F}$ such that:*

$$f(x) = \begin{cases} 0 & \text{if } x \in \mathcal{S}' \\ 1 & \text{if } x \in \{\mathcal{S} - \mathcal{S}'\} \end{cases} \tag{A.3}$$

Hence, a set $\mathcal{S}$ is shattered by a class of functions $\mathcal{F}$ if, for each labeling combination of elements within $\mathcal{S}$, there exists a function $f \in \mathcal{F}$ able to discriminate the positive and negative examples.

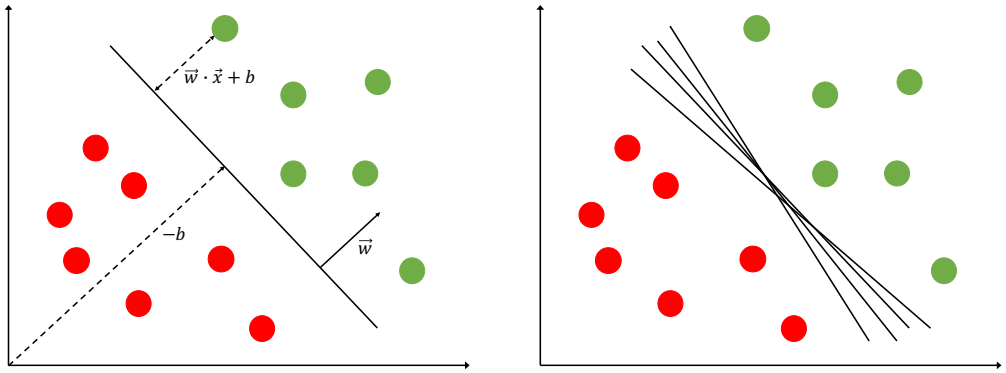
Definition A.3 (VC-dimension)**.** *The VC-dimension of a class of functions $\mathcal{F}$ is the maximum number of points that can be arranged so that $\mathcal{F}$ shatters them.*

Figure A.2 provides a geometrical explanation of the Definition A.3. When considering the class of linear functions, in a $\mathbb{R}^2$ space, 3 points can be always shattered, even selecting every possible configuration. This is not true when 4 points are considered, as for some configurations the points cannot be separated. Hence, in this case, $VC = 3$. It is worth noting that this upper-bound is not the same for every $\mathcal{F}$. For instance, if we consider the class of axis-aligned rectangles, it is easy to verify that there exist subsets of 4 points (randomly placed in the space) that can be shattered by this class, but, for every subset of 5 points, there are some classifications that cannot be attained. Therefore, here $VC = 4$.

The following Theorem correlates the error approximation in Definition A.1 with the *VC-dimension*.

Theorem A.1 (Vapnik and Chervonenkis)**.** *Let $\mathcal{H}$ be the hypothesis space and let be its VC-dimension equals to d. Let $\mathcal{S} \subseteq \mathcal{X}$ be a sample of m examples. Let D be a probability distribution over $\mathcal{X} \times \{-1, 1\}$. Then, $\forall h \in \mathcal{H}$, if $d \leq m$ and $m \geq \frac{2}{\epsilon}$:*

$$P(Err(h) \leq \epsilon(m, \mathcal{H}, \delta)) = 1 - \delta \tag{A.4}$$



(a) A separating hyperplane (b) Possible separating hyperplanes

Figure A.3. SVMs' hyperplanes

where

$$\epsilon(m, \mathcal{H}, \delta) = \frac{2}{m}\left(d \times \ln\left(\frac{2\epsilon \times m}{d}\right) + \ln\left(\frac{2}{\delta}\right)\right) \tag{A.5}$$

The above Theorem represents an essential building block for the Support Vector Machine (SVM) learning theory, that aims at minimizing the training set error, with a *VC-dimension* as low as possible.

The Support Vector Machine algorithm. Intuitively, the *VC-dimension* measures the complexity of the classifiers in the training examples set. When classifiers are simple enough, then their *VC-dimension* will be lower, thus minimizing the risk of error classification of unseen data.

One of the simplest classifier is a linear function:

$$f(\vec{x}, \theta) = sgn(\vec{w} \cdot \vec{x} + b) \tag{A.6}$$

The θ parameters of the above function correspond to (i) the gradient $\vec{w}$ of an hyperplane and (ii) its orthogonal distance b from the origin[1]. Instances are synthesized in *feature vectors* $\vec{x}$ in the geometrical space $\mathbb{R}^d$, where each dimension corresponds to a specific *feature* (or, aspect) of the observed example. Figure A.3(a) shows a $\mathbb{R}^2$ space where points represent training examples. In this vector space, examples are modeled considering just two features. Color represents the membership of each instance to the target class, being the green points examples of the target class. The aim of the learning algorithm is to acquire the parameters $\theta = \langle \vec{w}, b \rangle$ that allow to define a linear classifier separating all training examples. The distance of each point from the hyperplane, along with the sign of such value $sgn(\vec{w} \cdot \vec{x} + b)$, suggests on which side of hyperplane the examples lie. Hence, this value allows to determine, or predict, the class assignments. Such values will be positive for the examples belonging to the class, negative on the contrary.

The use of such classifiers in machine learning can be traced back to Rosenblatt's work on the perceptron algorithm [137]. This learning algorithm is fairly straightforward: instances are processed individually, and their class is predicted. If the output

[1] $\vec{w} \cdot \vec{x}$ denotes the inner product of two vectors

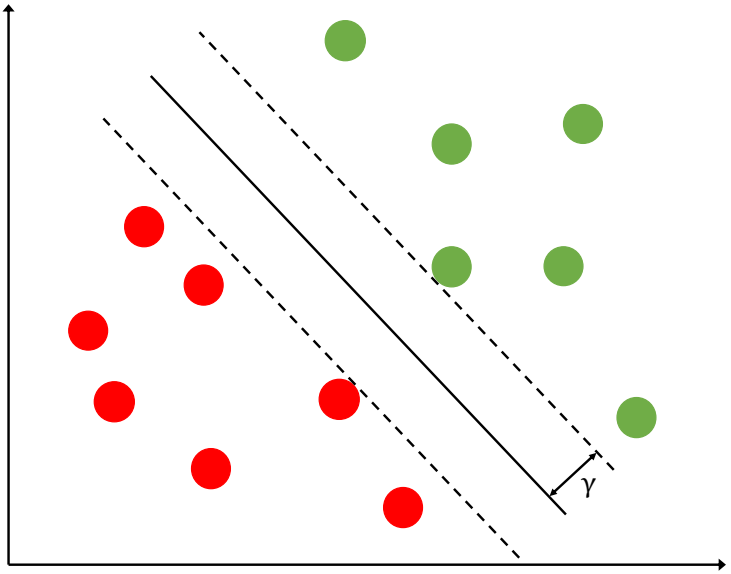


Figure A.4. Best separating hyperplane

prediction is correct, no adjustment is made; otherwise, the parameters $\vec{w}$ and b are moved in the direction of the point where the mistake occurred. The convergence of the algorithm after a finite number of iterations has been theoretically proved in [125]. However, such convergence is guaranteed only if the data set is linearly separable (see Definition A.4).

Definition A.4 (Linearly separable set)**.** *A set of points* $\{\vec{x}_i, y_i\}, i = 1, \ldots, l$ *where* $y_i \in \{-1, +1\}$ *are class labels, is called linearly separable if there exists a linear function* $f(\vec{x}_i)$ *such that:*

$$y_i \cdot f(\vec{x}_i) > 0 \quad \forall i = 1, \ldots, l \tag{A.7}$$

However, even if the resulting f allows to separate the training examples, this might not be true for unseen data. In fact, as shown in Figure A.3(b), the separating algorithm is not unique and the perceptron algorithm allows to learn only one classifier among all the possible existing. Thus, according to Theorem A.1, in order to learn the linear classifier that minimizes the classification error even over unseen data, the best function will be the one with the smallest value of *VC-dimension*.

A hyperplane is said to be *γ-margin separating hyperplane* if

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq \gamma, \forall i \tag{A.8}$$

where γ is the margin, as the geometric distance of an example $\vec{x}_i$ from the hyperplane (see Figure A.4). From a geometrical point of view, given the separating hyperplane, two parallel hyperplanes can be selected in a way that there are no training points between them. Hence, the distance between these two hyperplanes will be $2 \times \gamma$.

As we would want to prevent data points from falling within the margin region, a set of constraints are added to the targeted learning algorithm. In fact, the space can be linearly scaled for $||\vec{w}||$ such that:

$$y_i(\vec{w}^* \cdot \vec{x}_i + b^*) \geq 1, \forall i \tag{A.9}$$

where $\vec{w}^* = \frac{\vec{w}}{||\vec{w}||}$ and $b^* = \frac{b}{||\vec{w}||}$. We thus obtain a γ-margin separating hyperplane with $\gamma = \frac{1}{||\vec{w}||}$. The measure of the *VC-dimension* of the linear classifier expressed in Equation A.9 has a bound that has been defined through the following theorem [175]:

Theorem A.2. *Let $x \in \mathbb{R}^d$ be examples that belong to hyper-sphere of radius R. The set of γ-margin separating hyperplanes has VC-dimension h bounded by*

$$h \leq min\left(\frac{R^2}{\gamma^2}, d\right) + 1 = min\left(R^2||\vec{w}||^2, d\right) + 1 \tag{A.10}$$

Hence, among all possible linear classifiers, given a training set projected in a $\mathbb{R}^d$ space (with maximum *VC-dimension* $h = d + 1$), we can choose the one that minimizes the probability of misclassifying an unseen instance. Such optimal classifier is the one that has the lowest *VC-dimension*, i.e., the one maximizing the margin γ or, in other words, the one minimizing the gradient norm $||\vec{w}||$ of the hyperplane.

By considering the constraints imposed by Equation A.9, the max-margin separating hyperplane can be found by solving the following optimization problem, known as primal problem:

$$\begin{aligned} &minimize && \frac{1}{2}||\vec{w}||^2 \\ &subject\ to && y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1, \forall i = 1, \ldots, l \end{aligned} \tag{A.11}$$

A possible approach [25, 175] for solving this particular optimization problem introduces Lagrange multipliers α_i for each constraint. The resulting equation is known as the Lagrangian function:

$$L(\vec{w}, b, \alpha) = \frac{1}{2}||\vec{w}||^2 - \sum_{i=1}^{l} \alpha_i [y_i(\vec{w} \cdot \vec{x}_i + b) - 1] \tag{A.12}$$

In order to find the minimum of such function, derivatives are taken with respect to $\vec{w}$ and b, resulting in:

$$\vec{w} = \sum_{i=1}^{l} y_i \alpha_i \vec{x}_i \tag{A.13}$$

and

$$\sum_{i=1}^{l} y_i \alpha_i = 0 \tag{A.14}$$

It is worth noting that Equation A.13 defines that w is obtained by a specific linear combination of the training points. By replacing them into the primal problem expressed in Equation A.12, we obtain a dual formulation that allows to determine the α_i:

$$\begin{aligned} &maximize && W(\alpha) = \sum_{i=1}^{l} \alpha_i - \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \vec{x}_i \vec{x}_j \\ &subject\ to && \sum_{i=1}^{l} y_i \alpha_i = 0, \alpha_i \geq 0, \forall i = 1, \ldots, l \end{aligned} \tag{A.15}$$

The solution of such optimization problem is the set of α_i^* parameters optimizing $W(\alpha)$. As described in Equation A.13, the parameter hyperplane is expressed as the linear combination of training examples, $\vec{w}^* = \sum_{i=1}^{l} y_i \alpha_i^* \cdot \vec{x}_i$, and the corresponding margin is $\gamma^* = \frac{1}{||\vec{w}^*||}$. If $\alpha_i^* = 0$ for a given x_i, then the example is not used in the decision rule and can be discarded. Points x_i, such that $\alpha_i^* \neq 0$, lie on the margin and are called *support vectors*. Support vectors determine the decision boundary. The parameter b is not present in the dual problem, but it can be computed from any of the primal constraints in Equation A.9 as:

$$b^* = y_k - w^* \cdot x_k \quad \forall k : \alpha_k \neq 0 \tag{A.16}$$

The resulting classification function $f(\vec{x}, \vec{w}, b)$, or simply $f(\vec{x})$, can be thus expressed as:

$$f(\vec{x}, \vec{w}, b) = sgn(\vec{w} \cdot \vec{x} + b) = sgn\left(\sum_{i=1}^{l} y_i \alpha_i \vec{x}_i \cdot \vec{x} + b\right) \tag{A.17}$$

Such formulation can be applied to any linearly-separable case providing the so-called hard margin classifier, i.e., the hyperplane separating all examples without exception. However, it is not always a good solution forcing the classifier to separate all training examples, for the following reasons. First, examples are not always linearly separable. Second, some of the training examples could be accidentally assigned to the wrong class or some dimensions of the feature vector $\vec{x}_i$ could represent mis-information. Finally, some examples could be outliers, i.e., observations geometrically distant from the rest of the data belonging to the same class. In such cases a better solution might be to relax, when it is necessary, the constraints introduced in Equation A.9 as follows:

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 - \xi_i \tag{A.18}$$

The underneath idea is that points lying on the wrong side of the hyperplane are explicitly penalized by introducing slack variables ξ_i that control how far from the hyperplane a point can lies. To this end, the optimization problem becomes:

$$\begin{aligned} &\textit{minimize} && \frac{1}{2}||\vec{w}||^2 + C\sum_{i=1}^{l} \xi_i \\ &\textit{subject to} && y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0, \forall i = 1, \ldots, l \end{aligned} \tag{A.19}$$

where the parameter C is a regularization term, which provides a way to control the trade-off between the size of the margin and cumulative training error and, therefore, to prevent overfitting. C is chosen by the user, even though are no general methods for choosing it value; conversely, it is usually set to optimize some performance measure on the training or validation set during the so-called tuning process. This particular formulation of the optimization problem is called a *soft-margin* classification. As C becomes large, it is unattractive to not respect the data at the cost of reducing the geometric margin: the result is that Equation A.18 will tend to approximate the hard-margin definition of Equation A.11. Conversely, when it is small, it is easy to account for some data points with the use of slack variables

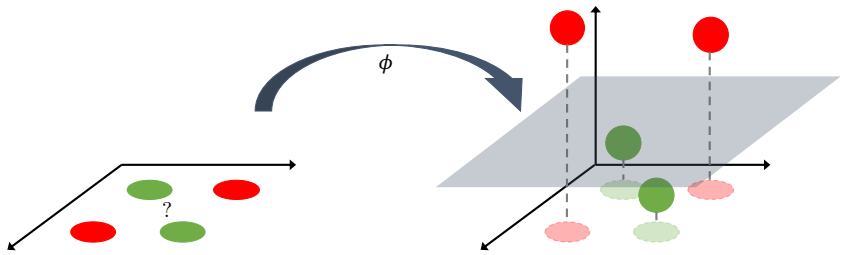


Figure A.5. A mapping ϕ which makes separable the initial data points

and to have a fat margin placed, so that it models the bulk of the data. Again, the solution is then given by a linear combination of inner products with support vectors.

Kernel methods. As discussed in the previous paragraph, SVMs learn linear classifiers that are, geometrically speaking, a separating hyperplane $f(\vec{x}) = \vec{w} \cdot \vec{x} + b = 0$, where $\vec{x}$ is the feature vector representation of a classifying object and $\vec{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are the acquired parameters.

However, such a linear classifier is not always learnable from training data, as the training examples might not be linearly separable. The soft-margin formulation expressed in Equation A.19 aims at alleviate such problem by allowing the algorithm to ignore some training instances, that would compromise the quality of the resulting generalization. When this solution is not applicable, as the non-linearly separation is a property of the data distribution rather than few isolated instances, a more suitable approach might be required.

On the left of Figure A.5, a $\mathbb{R}^2$ space where objects are not linearly separable is shown. A straightforward solution might be to define a higher complexity classification function, consequently characterized by a higher value of *VC-dimension*. An undesired side-effect of this approach is that the risk of misclassification of unseen data increases. Another solution consists in increasing the vector space dimensionality, by adding novel synthetic dimensions, as shown in Figure A.5 on the right. In such a $\mathbb{R}^3$ space a more informative observation would provide a more representative space, where a linear classifier (here the bi-dimensional plane) could be easily learned. This corresponds to the solution provided by manual features engineering. Tough it might actually provide an effective improvement of the resulting model, it is not a feasible approach in most cases.

Conversely, a viable solution is to define an effective function, allowing to improve the representation of training examples without providing an explicit feature engineering step. Such a conversion process is modeled in terms of a projection function $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$, mapping points $\vec{x}$ from a d-dimensional space to a d'-dimensional space, through the application of $\phi(\vec{x}_i)$. In the resulting space, the learning algorithm can be then applied, so that the classification function in Equation A.17 becomes:

$$f(\vec{x}, \vec{w}, b) = sgn(\vec{w} \cdot \phi(\vec{x}) + b) = sgn\left(\sum_{i=1}^{l} y_i \alpha_i \phi(\vec{x}_i) \cdot \phi(\vec{x}) + b\right) \tag{A.20}$$

Moreover, the dual formulation of SVM in Equation A.15 suggests that learning

does not depend on the $\vec{x}_i$ geometric representations but only on their pairwise dot products. Hence, the projection function itself is not actually essential. Hence, the problem can be reduced to the computation of a Kernel function $K(\vec{x}_i, \vec{x}_j) = \phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$, so that the optimization problem can be rewritten as:

$$\begin{aligned} &maximize && W(\alpha) = \sum_{i=1}^{l} \alpha_i - \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j K(\vec{x}_i, \vec{x}_j) \\ &subject\ to && \sum_{i=1}^{l} y_i \alpha_i = 0, \alpha_i \geq 0, \forall i = 1, \ldots, l \end{aligned} \tag{A.21}$$

while the classification function is expressed as:

$$f(\vec{x}, \vec{w}, b) = sgn\left(\sum_{i=1}^{l} y_i \alpha_i \phi(\vec{x}_i) \cdot \phi(\vec{x}) + b\right) = sgn\left(\sum_{i=1}^{l} y_i \alpha_i K(\vec{x}_i, \vec{x}_j) + b\right) \tag{A.22}$$

This is known as the *"kernel trick"*, as the kernel function is directly applied over data without ever using the mapping ϕ. During tagging, the hyperplane is not directly defined, as it is the linear combination of support vectors, but the classification is still feasible in terms of the similarity (the dot-product) among the novel instances and the support vectors. The explicit representation of the novel feature space is thus never built and it is thus called *implicit feature space*.

Kernel Methods [150] refer to a large class of learning algorithms based on inner product vector spaces, among which SVMs are one of the most well known learning algorithms. The learning algorithm will select the most representative instances and features in the implicit space, i.e., the space dimensions. Such methods provide effective statistical predictions without focusing over the construction of *ad-hoc* feature representations, but defining meaningful similarity (i.e., kernels) functions among examples. Moreover, Kernel methods have the advantage that linear combinations of kernel functions, such as kernel sum or product, can be easily integrated into SVMs as, in line with [150], such combinations are considered Kernels themselves. The choice of the kernel combination strategy can be also based on prior knowledge about the problem. These combinations are very useful to mix the information provided by the original features, for example acting on different perspectives (e.g., lexical vs. syntagmatic properties of a sentence or text) on the original objects, e.g., textual units.

Markovian Model as Classifiers

A generative model assumes the existence of a probability distribution generating data. However, by forcing the algorithm to estimate the distribution function and its unknown parameters from observations, a generative model is able to predict labels for novel examples. However, such a behavior can be achieved only taking into account some preliminary assumptions:

- data are *independent and identically distributed* (*i.i.d.* property);
- data are generated from a *mixture model*;

- there exists a one-to-one correspondence between *mixture component* and label classes.

As mixture models are able to represent data through different probability distributions, a generative classification task aims at estimating the distribution that generated a novel example.

Markov chain. A Markov chain is a stochastic process with the *Markov property*. The term *"Markov chain"* refers to the sequence of random variables such a process moves through, with the Markov property defining serial dependence only between adjacent periods (as in a "chain"). Hence, it can be used for describing systems that follow a chain of linked events, where what happens next depends only on the current state of the system, and not on the past states.

Definition A.5 (Markov chain)**.** *A Markov chain is a sequence of random variables* $X_1, X_2, X_3, \ldots$ *where the following properties hold:*

- ***Limited Horizon Property*** *(i.e. Markov property)*
  $P(X_{n+1} = x_k | X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = P(X_{n+1} = x_k | X_n = x_n)$
- ***Time Invariant Property***
  $P(X_{n+1} = x_k | X_n = x_l) = P(X_2 = x_k | X_1 = x_l)$

The possible values of X_i *form a countable set* S *called the state space of the chain.*

Moves from a state to another one of the system are called **transitions** and the probabilities $p_{l,k} = P(X_{n+1} = x_k | X_n = x_l)$ associated to the different transitions are called **transition probabilities**. Therefore, the process is characterized by:

- a state space $S = \{x_1 \ldots x_n\}$
- a stochastic[2] transition matrix P enumerating the transition probabilities

$$P = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,1} & p_{n,2} & \cdots & p_{n,n} \end{pmatrix}$$

- an initial distribution across the state space

$$\pi_i = P(X_1 = x_i)$$

In Figure A.6, an example of Markov chain is provided. The transition probabilities correspond to the following transition matrix:

$$P = \begin{pmatrix} 0.4 & 0.25 & 0 & 0 & 0.35 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0.25 & 0 & 0.75 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 & 0 \\ 0.9 & 0 & 0 & 0.1 & 0 & 0 \end{pmatrix}$$

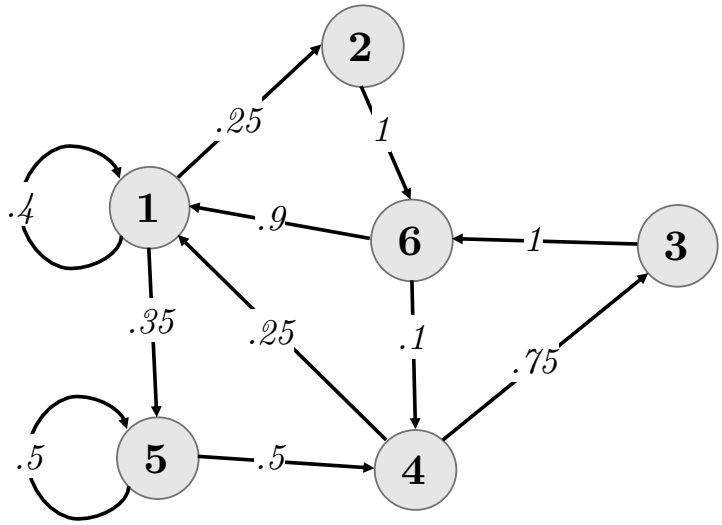


Figure A.6. Example of Markov chain (automa representation)

By convention, we assume that all states and transitions have been included in the definition of the process, so that there is always a next state, and the process does not terminate.

Hidden Markov Model. A Hidden Markov Model (HMM) is a statistical Markov model where the system being modeled is assumed to be a Markov process with unobserved (or, *hidden*) states. In fact, given the current state, an outcome (or observation) is generated, according to:

- the associated probability distribution, i.e. *emission probability*, and
- the transition probability from a state to another, i.e. *transition probability*.

A formal definition is provided in the following.

Definition A.6 (Hidden Markov Model)**.** *A HMM is a tuple* $< S, O, P, B, \vec{\pi} >$*, where:*

- $S = \{x_1 \dots x_n\}$ *is the state space;*
- $O = \{o_1 \dots o_m\}$ *is the output symbols space;*
- P *is a stochastic transition matrix that describes the probabilities of particular transitions (i.e., transition probabilities)*

$$P = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,1} & p_{n,2} & \cdots & p_{n,n} \end{pmatrix}$$

[2]Note that, a matrix P is stochastic iff $\forall i, j \quad p_{i,j} \geq 0$ and $\sum_{j=1}^{n} p_{i,j} = 1$

- *B is an emission matrix such that, for each state x_i and for each possible output o_j, $b_i(o_j)$ (or, alternatively, $b_{i,j}$) is the probability that a particular output symbol o_j is observed in a state x_i; in other words, $b_{i,j}$ gives the probability that o_j is emitted in state x_i.*

$$B = \begin{pmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,n} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n,1} & b_{n,2} & \cdots & b_{n,n} \end{pmatrix}$$

  The elements of B matrix are called emission probabilities;

- *$\vec{\pi}$ is the initial distribution across the state space, where each π_i is*

$$\pi_i = P(X_1 = x_i)$$

In general, when dealing with HMM, three different problems have to be taken into account:

- *Likelihood*: compute the probability of an output sequence $(o_1, \ldots, o_n)$, given a model $< S, O, P, B, \vec{\pi} >$;
- *Decoding*: given a model, compute the most likely state sequence that generates an observed output sequence;
- *Parameter estimation*: given a set of examples of output sequence and a model space, find the most likely model that generates the example set.

The above problems are further analyzed in the following.

Likelihood. The *Likelihood* problem [92] is the task of computing, given the parameters of the model, the probability of a particular output sequence $P(O_i)$, where $O_i = (o_{k_1}, \ldots, o_{k_T})$ is an observed sequence. Among the possible solutions, one is by applying *brute force* as follows: compute $P(O_i)$ by summarizing the probabilities of all paths $(s_{i_1}, \ldots, s_{i_T})$ that are able to generate O_i. However, this approach is not feasible, as it can be performed at high computational cost.

A more viable approach is to apply the principle of *dynamic programming*. The idea behind dynamic programming is quite simple. In general, a given problem is decomposed into different subproblems, their solutions are stored and then combined to get an overall solution. It can be seen as a further optimization of the *divide and conquer* approach, where, due to recursion, many of the subproblems might be generated and solved many times. Conversely, the dynamic programming method seeks to solve each subproblem only once, thus reducing the number of computations: whenever the solution to a given subproblem has been computed, it is stored and looked-up when needed, without any other computational overhead. The Likelihood problem is usually handled through the *Forward Algorithm*, that is an instance of the dynamic programming pattern.

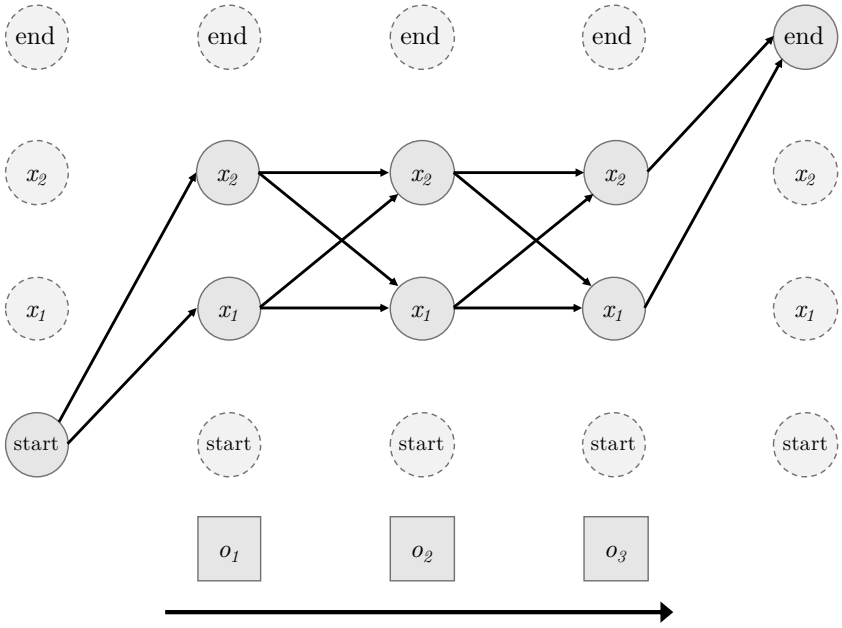


Figure A.7. Example of *trellis*

Let Θ_t be a random variable that represents the output symbol at time t. Let $\alpha_t(i)$ be the probability of the partial observation sequence $O_{\leq t} = (o_{k_1}, o_{k_2}, \ldots, o_{k_t})$ to be produced by all possible state sequences that end at the i^{th} state:

$$\alpha_t(i) = P(O_{\leq t}; X_t = x_i) = P(\Theta_1 = o_{k_1}, \ldots, \Theta_t = o_{k_t}; X_t = x_i) \tag{A.23}$$

Then, the unconditional probability of the partial observation sequence is the sum of $\alpha_t(i)$ over all N states. Forward Algorithm is a recursive algorithm for calculating $\alpha_t(i)$ for the observation sequence of increasing length t. First, the probabilities for the single-symbol sequence are calculated as a product of initial i^{th} state probability and emission probability of the given symbol o_{k_1} in the i^{th} state:

$$\alpha_1(i) = \pi_i \cdot b_i(o_{k_1}) \tag{A.24}$$

Then, the recursive formula is applied. Assume $\alpha_{t-1}(i)$ has been calculated for some $t-1$. To calculate $\alpha_t(j)$, every $\alpha_{t-1}(i)$ is multiplied by the corresponding transition probability $b_{i,j}$ from the i^{th} state to the j^{th} state, sum the products over all states, and then multiply the result by the emission probability of the symbol o_{k_t}, $b_i(o_{k_t})$:

$$\alpha_t(i) = \left[\sum_{j=1}^{N} \alpha_{t-1}(j) p_{i,j}\right] \cdot b_i(o_{k_t}) \tag{A.25}$$

Iterating the process, $\alpha_T(i)$ can be calculated, and then summing them over all states, the required probability is obtained.

Figure A.7 shows the *trellis diagram* that allows to efficiently represent the Forward Algorithm.

Decoding. As well as the Likelihood one, the *Decoding problem* [92] is solved through dynamic programming, as brute force would result too much expensive in terms of computational cost.

The *Viterbi algorithm* is a dynamic programming approach that is usually used to solve the Decoding problem. It chooses the best path that maximizes the likelihood of the state sequence for a given observation sequence. The Viterbi Algorithm uses the same schema as the Forward algorithm, but it employs maximization in place of summation within the recursion step. Let $\delta_t(i)$ be the maximal probability of state sequences of the length t that end in state i and produce the t first observations for the given model. Then, the probabilities for the single-symbol sequence are calculated as:

$$\delta_1(i) = \pi_i \cdot b_i(o_{k_1}) \tag{A.26}$$

whereas each $\delta_t(i)$ is computed as:

$$\delta_t(i) = max_{1 \leq j < N} \left[\delta_{t-1}(j) p_{i,j}\right] \cdot b_i(o_{k_t}) \tag{A.27}$$

Parameter estimation. The *Parameter Estimation* problem refers to the computation of the model parameters, namely the transition probabilities matrix P, the emission probabilities matrix B and the initial distribution $\vec{\pi}$.

Several supervised approaches allow to compute, given a training set of labeled examples, the above parameters through maximum likelihood estimation, by exploiting the evidences of the frequencies of the observation. Other approaches aim at iteratively improving the estimation of parameters [92]. These methods are known as Expectation Maximization (EM) algorithms [107].

HMMs are the most popular techniques of temporal classification, finding application in manifold areas like speech, handwriting and gesture recognition. In the textual domain, these models are often employed in Named Entity recognition and Part-Of-Speech (POS) tagging tasks.

Structural Support Vector Machines

A different formulation of the HMM optimization problem has been proposed in [165]. In contrast with the SVM algorithm introduced in Section A.1.1, given a set of pairs $(\vec{x}_1, \vec{y}_1), \ldots, (\vec{x}_n, \vec{y}_n) \in \mathcal{X} \times \mathcal{Y}$, Structural Support Vector Machine (SVM^{struct}) learns a function $f : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{Y}$ is a set of structured outputs, such as sequences, sets, or trees. The approach is to learn a *discriminant function* $F : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ over input/output pairs from which a prediction is derived by maximizing F over the response variable for a specific given input $\vec{x}$. Hence, the general form of hypothesis f is

$$f(\vec{x}; \vec{w}) = \arg\max_{\vec{y} \in \mathcal{Y}} F(\vec{x}, \vec{y}; \vec{w}) \tag{A.28}$$

where $\vec{w}$ denotes a parameter vector. It might be useful to think of F as a $\vec{w}$-parameterized family of cost functions. F is assumed to be linear in some *combined feature representation* of inputs and outputs $\Psi(\vec{x}, \vec{y})$,

$$F(\vec{x}, \vec{y}; \vec{w}) = \langle \vec{w}, \Psi(\vec{x}, \vec{y}) \rangle \tag{A.29}$$

where $\Psi(\vec{x}, \vec{y})$ depends on the nature of the specific problem.

Learning over structured output spaces $\mathcal{Y}$ inevitably involves loss functions other than the standard zero-one classification loss: it is assumed the availability of a bounded loss function $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ where $\Delta(\vec{y}, \vec{y}^*)$ quantifies the loss associated with a prediction $\vec{y}^*$, if the true output value is $\vec{y}$.

Assuming that $\Delta(\vec{y}, \vec{y}^*) > 0$ for $\vec{y} \neq \vec{y}^*$ and $\Delta(\vec{y}, \vec{y})$, then the condition of zero training error can then be compactly written as a set of non-linear constraints

$$\forall i : \max_{\vec{y} \in \mathcal{Y} \backslash \vec{y}_i} \langle \vec{w}, \Psi(\vec{x}_i, \vec{y}) \rangle \leq \langle \vec{w}, \Psi(\vec{x}_i, \vec{y}_i) \rangle \tag{A.30}$$

Each nonlinear inequalities in A.30 can be equivalently replaced by $|\mathcal{Y}| - 1$ linear inequalities, resulting in a total of $n|\mathcal{Y}| - n$ linear constraints,

$$\forall i, \forall \vec{y} \in \mathcal{Y} \backslash \vec{y}_i : \langle \vec{w}, \delta\Psi_i(\vec{y}) \rangle > 0 \tag{A.31}$$

where $\delta\Psi_i(\vec{y}) \equiv \Psi(\vec{x}_i, \vec{y}_i) - \Psi(\vec{x}_i, \vec{y})$.

The resulting *hard-margin* optimization problem is

$$\begin{cases} \min\limits_{\vec{w}} \frac{1}{2}||\vec{w}||^2 \\ \forall i, \forall \vec{y} \in \mathcal{Y} \backslash \vec{y}_i : \langle \vec{w}, \delta\Psi_i(\vec{y}) \rangle \leq 1 \end{cases} \tag{A.32}$$

To allow errors in the training set, slack variables are introduced to optimize a soft-margin criterion:

$$\begin{cases} \min\limits_{\vec{w}, \xi} \ \frac{1}{2}||\vec{w}||^2 + \frac{C}{n} \sum\limits_{i=1}^{n} \xi_i \ s.t. \forall i, \xi_i \geq 0 \\ \forall i, \forall \vec{y} \in \mathcal{Y} \backslash \vec{y}_i : \langle \vec{w}, \delta\Psi_i(\vec{y}) \rangle \leq 1 - \xi_i \end{cases} \tag{A.33}$$

where $C > 0$ is a constant that controls the trade-off between training error minimization and margin maximization.

The Equation A.33 implicitly considers the zero-one classification loss; this is inappropriate for problems like natural language parsing, where $|\mathcal{Y}|$ is large. An approach is to *re-scale* the slack variables according to the loss incurred in each of the linear constraints. Intuitively, violating a margin constraint involving a $\vec{y} \neq \vec{y}_i$ with high loss $\Delta(\vec{y}_i, \vec{y})$ should be penalized more severely than a violation involving an output value with smaller loss. This can be accomplished by multiplying the violation by the loss, or equivalently, by scaling slack variables with the inverse loss:

$$\begin{cases} \min\limits_{\vec{w}, \xi} \ \frac{1}{2}||\vec{w}||^2 + \frac{C}{n} \sum\limits_{i=1}^{n} \xi_i \ s.t. \forall i, \xi_i \geq 0 \\ \forall i, \forall \vec{y} \in \mathcal{Y} \backslash \vec{y}_i : \langle \vec{w}, \delta\Psi_i(\vec{y}) \rangle \leq 1 - \frac{\xi_i}{\Delta(\vec{y}_i, \vec{y})} \end{cases} \tag{A.34}$$

The dual formulation of Equation A.32 can be derived as in normal SVM. Let $\alpha_{i\vec{y}}$ the Lagrange multiplier enforcing the margin constraint for label $\vec{y} \neq \vec{y}_i$ and example $(\vec{x}_i, \vec{y}_i)$. Using standard Lagrangian duality techniques, one arrives at the following dual formulation

$$\begin{cases} \max\limits_{\alpha} \sum\limits_{i, \vec{y} \neq \vec{y}_i} \alpha_{i\vec{y}} - \frac{1}{2} \sum\limits_{\substack{i, \vec{y} \neq \vec{y}_i \\ j, \vec{y} \neq \vec{y}_j}} \alpha_{i\vec{y}} \alpha_{j\vec{y}} \langle \delta\Psi_i(\vec{y}), \delta\Psi_j(\vec{y}^*) \rangle \\ s.t. \ \forall i, \forall \vec{y} \in \mathcal{Y} \backslash \vec{y}_i : \alpha_{i\vec{y}} \geq 0 \end{cases} \tag{A.35}$$

Examples of problems with complex outputs are natural language parsing, sequence alignment in protein homology detection, and Markov models for part-of-speech tagging. Moreover, the SVM^{struct} algorithm can also be used for linear-time training of binary and multi-class SVMs.

Multi-Class Support Vector Machine classifier. The *multi-classification* problem refers to a classification task where the labels set cardinality is $|\mathcal{Y}| \geq 2$. A possible approach is to exploit several binary classifiers in a *One-VS-All* fashion: for each class $y \in \mathcal{Y}$ a binary classifier is trained and the resulting label y^* is the one with the best predicting score. This approach is quite complex, as $|\mathcal{Y}|$ training and classification steps are required.

The maximization function is defined as:

$$\begin{cases} min\ \frac{1}{2}\sum_{i=1}^{k}|w|^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i \\ \\ s.t.: \\ \ \forall y \in [1,\ldots,k] : [x_1 \cdot w_{y_i}] \geq [x_1 \cdot w_y] + 100 \cdot \Delta(y_1, y) - \xi_1 \\ \ldots \\ \ \forall y \in [1,\ldots,k] : [x_n \cdot w_{y_n}] \geq [x_n \cdot w_y] + 100 \cdot \Delta(y_n, y) - \xi_n \end{cases} \tag{A.36}$$

The above optimization problem is very fast in the linear case; moreover, it enables the *Winner-Takes-All* multi-class classification [39].

Hidden Markov Support Vector Machines. The Hidden Markov Support Vector Machine (SVM^{hmm}) classifier implements a structured SVM that is able to predict labels sequences. This formulation [5] refers to the HMM theoretical constructs introduced in Section A.1.1.

Given an input sequence of feature vectors $X = (\vec{x}_1, \ldots, \vec{x}_l)$, the model predicts a label sequence $Y = (y_1, \ldots, y_l)$, according to the linear discriminant function:

$$Y^* = \arg\max_{\forall Y} \left\{ \sum_{i=1}^{l} \left[\sum_{j=1}^{k} (\vec{x}_i \cdot \vec{w}_{y_{i-j}\ \ldots\ y_i}) \ +\ \Phi_{tr}(y_{i-j},\ \ldots, y_i) \cdot \vec{w}_{trans} \right] \right\} \tag{A.37}$$

where

- $\vec{w}_{y_{i-k}\ \ldots\ y_i}$ is the emission weight vector for the k^{th}-order label sequence $(y_{i-k}\ \ldots\ y_i)$;
- $\Phi_{tr}(y_{i-k},\ \ldots, y_i)$ is an indicator vector that has exactly one entry set to 1 corresponding to the sequence $(y_{i-k}\ \ldots\ y_i)$;
- $\vec{w}_{trans}$ is a transition weight vector for the transition weight between adjacent labels.

In line with the structured formulation, the SVM^{hmm} classifier requires the implementation of a function $\Psi(X, Y)$ that considers two type of features:

- the interactions between attributes of the observation $\vec{x}_i$ and a specific label y_i, i.e. the $\vec{x}_i$ by y_i emission property;

- interactions between neighboring labels y_i along the chain.

In order to assign a label sequence to an input chain, the function $\Psi(X, Y)$ has been developed so that a *Viterbi decoding algorithm* can be applied.

Given a set of pairs $(X^1, Y^1), \ldots, (X^n, Y^n)$, where each $X^k = (\vec{x}_1^k, \ldots, \vec{x}_l^k)$ is a sequence of vectors and each $Y^k = (y_1^k, \ldots, y_l^k)$ is the corresponding label sequence, a SVM^{hmm} classifier is trained solving the following optimization problem

$$\begin{cases} min\ \frac{1}{2}|\vec{w}|^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i \\ s.t. \\ \forall Y: \sum_{i=1}^{l}(\vec{x}_i^1 \cdot \vec{w}_{y_i^1}) + \Phi_{trans}(y_{i-1}^1, y_i^1)\cdot\vec{w}_{trans} \geq \\ \quad \sum_{i=1}^{l}(\vec{x}_i^1 \cdot \vec{w}_{y_i}) + \Phi_{trans}(y_{i-1}, y_i)\cdot\vec{w}_{trans} + \Delta(Y^1, Y) - \xi_1 \\ \ldots \\ \forall Y: \sum_{i=1}^{l}(\vec{x}_i^n \cdot \vec{w}_{y_i^n}) + \Phi_{trans}(y_{i-1}^n, y_i^n)\cdot\vec{w}_{trans} \geq \\ \quad \sum_{i=1}^{l}(\vec{x}_i^n \cdot \vec{w}_{y_i}) + \Phi_{trans}(y_{i-1}, y_i)\cdot\vec{w}_{trans} + \Delta(Y^n, Y) - \xi_n \end{cases} \tag{A.38}$$

where $\Delta(Y^i, Y)$ is the loss function, computed as the number of misclassified labels in the sequence.

The proposed SVM^{hmm} algorithm is parametric with respect to emission and transition orders. In other words, with a transition order $k > 1$, the training and classification tasks will depend on a k-transition order Markov chain. Note that the higher is the k value, the higher is the training computational time, as a longer story of transition is considered.

A.1.2. An Introduction to Automated Decision Making

Decision making is the cognitive process of making choices by identifying a decision, gathering information, and assessing alternative resolutions. Every process produces a decision, that is expected to be optimal given the gathered information. In literature, manifold mathematical frameworks have been proposed to model the decision making process in autonomous agents. This section introduces a class of stochastic decision-making algorithms, Markov Decision Processs (MDPs), whose main goal is to maximize the expected utility of a sequence of interactions with a stochastic process. Then, Reinforcement Learning (RL) is reviewed as an approach for solving a MDP by acquiring the required transition probabilities.

Markov Decision Processes

The MDP [20] is a well-known model designed for planning and decision making in discrete settings. In this framework, the decision process is modeled through a set of states $\mathcal{S}$ and a set of actions $\mathcal{A}$, enabling the agent to transition from a state $s \in \mathcal{S}$ to the next state $s' \in \mathcal{S}$. The overall process can be thus described as a graph, where nodes represent the states $\mathcal{S}$ and *edges* (or, arcs) refer to actions $\mathcal{A}$. Each pair $\langle s, a\rangle$ is associated to reward $r_{s,a}$; the transition from states to states can be

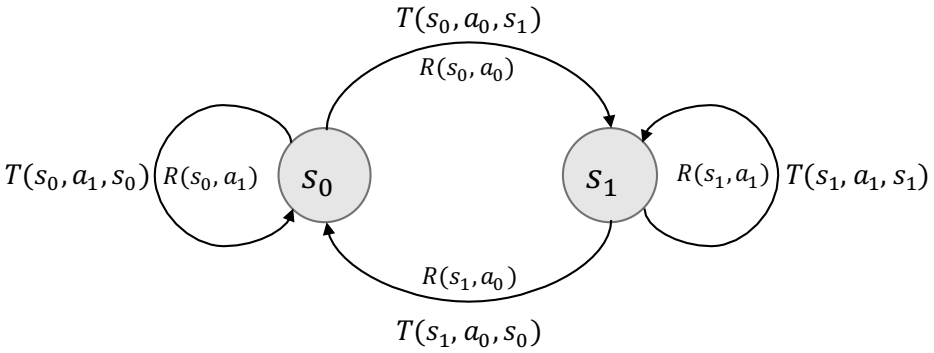


Figure A.8. Example of a Markov Decision Process

deterministic or stochastic. Depending on the ability of the agent to observe the entire current environmental state, the decision process might be called either *fully observable* (MDP) or *partially observable* (Partially Observable Markov Decision Process (POMDP)). For the purposes of this thesis, only the fully observable setting will be reviewed. A MDP can be thus formalized as follows:

Definition A.7 (Markov Decision Process)**.** *A **Markov decision process** is a tuple*

$$\mathcal{MDP} = (\mathcal{S}, \mathcal{A}, T, R, \gamma)$$

where:

- $\mathcal{S}$ *is the set of states of the environment;*
- $\mathcal{A}$ *represents the set of actions;*
- $T : S \times A \times S \to [0, 1]$ *is a probabilistic transition function modeling the transition from state* $s \in \mathcal{S}$ *to* $s' \in \mathcal{S}$*, when the agent takes the action* $a \in \mathcal{A}$*;*
- $R : S \times A \to \mathbb{R}$ *is the reward function, mapping a pair* $\langle s, a \rangle$ *into a real-valued reward* $r_{s,a} \in \mathbb{R}$*;*
- $\gamma \in (0, 1]$ *is a discount factor.*

Figure A.8 shows an example of MDP, where both the sets of states $\mathcal{S}$ and actions $\mathcal{A}$ are composed of two elements, $\{s_0, s_1\}$ and $\{a_0, a_1\}$, respectively. Transition probabilities and reward function can thus be represented by two 2×2 matrices, covering all the possible combinations of states/actions.

Given the current formalization, transitions and rewards are subject to the Markovian property, so that they depend just on the current state. Conversely, decisions are represented through a policy π, defining the *behavior* of the agent. In fact, π provides a mapping from states to actions. Policies can be either *deterministic* or *stochastic*. While in deterministic policies $\pi(s)$ the unique action is based only on the current state, stochastic policies provide a probability distribution $\pi(a|s) \in [0, 1]$ over the whole actions set $\mathcal{A}$.

When a policy is being executed, the agent interacts with its environment in discrete time-steps, defining a *sequence* of state-action pairs $\zeta = (\langle s_0, a_0\rangle, \ldots, \langle s_T, a_T\rangle)$, and the corresponding *cumulative reward* $R(\zeta) = \sum_{t=0}^{T} \gamma^t R(s_t, a_t)$. Hence, the goal of a deterministic agent is to find a policy $\pi(s)$, such that its *expected cumulative reward* $\mathbb{E}_{\zeta|\pi}[R(\zeta)]$ is maximized. The expected cumulative reward can be obtained through the *state-value function* $V^\pi(s)$:

$$V^\pi(s) = \sum_a \pi(a|s) Q^\pi(s,a) \tag{A.39}$$

where $Q^\pi(s,a)$ is the *action-value function*:

$$Q^\pi(s,a) = R(s,a) + \gamma \sum_{s'} T(s,a,s') V^\pi(s'). \tag{A.40}$$

Intuitively, while the state-value function expresses the expected value of following policy π forever when the agent starts following it from state s, the action-value function represents the expected value of first taking action a from state s and then following policy π forever.

In line with [157], under the Bellman's *Principle of Optimality* [21], where *"an optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision"*, it is possible apply the Bellman optimality equation to Eq. A.39 and A.40:

$$V^*(s) = \max_a \{R(s,a) + \gamma \sum_{s'} T(s,a,s') V^*(s')\} = \max_a Q^*(s,a) \tag{A.41}$$

$$\begin{aligned} Q^*(s,a) &= R(s,a) + \gamma \sum_{s'} T(s,a,s') \max_{a'} Q^*(s',a') \\ &= R(s,a) + \gamma \sum_{s'} T(s,a,s') V^*(s') \end{aligned} \tag{A.42}$$

obtaining the definition of an optimal policy greedily determined with one look-ahead, as in Eq. A.43, or just by choosing the best action according the optimal action-value function A.44:

$$\pi^*(s) = \arg\max_a \{R(s,a) + \gamma \sum_{s'} T(s,a,s') V^*(s')\} \tag{A.43}$$

$$\pi^*(s) = \arg\max_a \{Q^*(s,a)\}. \tag{A.44}$$

Learning Decision Policies

Once an environment has been defined as a MDP, it remains unsolved how to properly shape and acquire a policy, that allows an agent to navigate through the environment.

RL is an area of machine learning related to modeling the way agents can take actions within an environment, in order to maximize some notion of cumulative reward. Hence, a RL agent interacts with its MDP environment in discrete time steps. At each time t, the agent observes the current state s_t. Then, it chooses an action a_t from the set of available actions, which is subsequently reflected into the

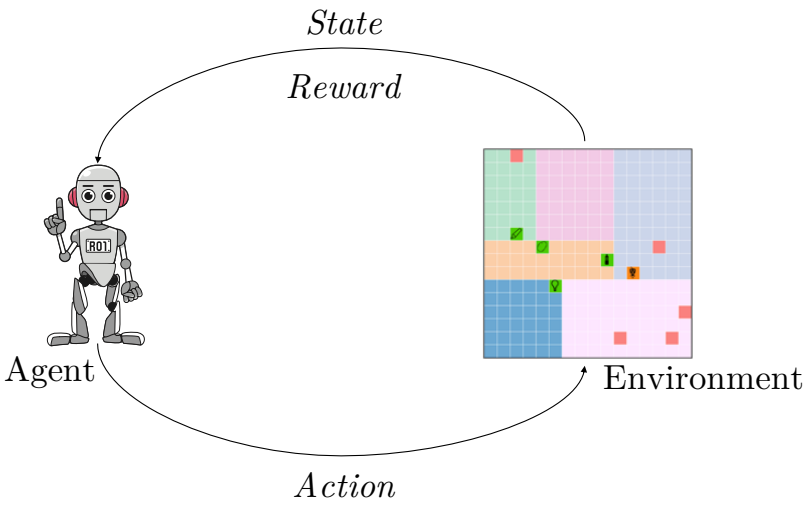


Figure A.9. Typical sketch of a Reinforcement Learning agent

environment. This decision produces a reward r_{s_t,a_t} for the agent. The environment thus moves to a new state s_{t+1}. At the end, the goal of a RL agent is to get as much reward as possible. In Figure A.9, a typical representation of the whole process is displayed.

The main difference between RL and standard supervised learning algorithms lies in how examples are presented to the learning algorithm. In fact, while standard supervised learning requires a dataset of labeled input/output pairs to acquire the model, RL doeas not acquire the policy starting from a labeled dataset. Conversely, RL leverages the concept of policy performance, finding a trade-off between exploration (of the uncharted state space) and exploitation (of the current policy). In fact, to better assess an optimal policy, a RL agent requires a proper tuning of the exploration mechanism. The random selection of actions results not the best choice in terms of performance. A solution is represented by the ϵ-greedy method: the agent chooses an action according to its policy π (the one providing the best long-term effect) with probability $1 - \epsilon$; otherwise, with probability ϵ an action is uniformly drawn from the set $\mathcal{A}$. It is worth noting that $0 < \epsilon < 1$ is a tuning parameter, which is either changed to make the agent explore progressively less, or based on some heuristics.

SARSA algorithm. The State-Action–Reward-State-Action (SARSA) is a RL algorithm for the acquisition of a MDP policy. It has been initially proposed by Rummery and Niranjan [141] with the name Modified Connectionist Q-Learning (MCQ-L) and then refined by Sutton and Barto [157]. SARSA owes its name to the way the Q-value is updated. In fact, the action-value function $Q(s_t, a_t)$ depends on the current state of the agent s_t (**S**), the action the agent chooses a_t (**A**), the reward r_{s_t,a_t} the agent gets for choosing this action (**R**), the state s_{t+1} that the agent goes given that action (**S**), and, finally, the next action a_{t+1} (**A**) the agent chooses in the new state s_{t+1}.

Operationally, the Q-value $Q(s_t, a_t)$ for the state-action pair $\langle s_t, a_t \rangle$ represents

the possible reward received in the next time step for taking action a_t in state s_t, plus the discounted future reward received from the next state-action observation, performed with a given learning rate. More formally:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right] \tag{A.45}$$

where α and γ are the hyperparameters. In fact, α controls the *learning rate*, determining to what extent newly acquired information overrides old ones. When $\alpha = 0$, the agent doe not learn anything, whereas with $\alpha = 1$ the agent consider only the most recent information. Conversely, γ is a discount factor defining to what extent future rewards impact on the Q-value. Again, when $\gamma = 0$, the agent considers only the current reward; on the contrary, if $\gamma \to 1$, long-term reward will highly impact on the Q-value.

A.1.3. Generalizing Lexical Semantics through Distributional Models

In Section A.1, ML has been introduced as a dichotomy between *Representation* and *Generalization*. The latter has been accounted into the previous sections, by introducing several ML approaches and algorithms. The final goal of these machines is to generalize a concept, starting from a bunch of examples represented in a machine-readable fashion. Into this section, we will explain the way such vectors are composed, with a special focus on the linguistic domain.

The representation of words and their meaning is a central problem in Computational Linguistics. When language learning is applied to generalize linguistic observations of the targeted phenomena, the information carried by single words play a crucial role in the resulting quality of the underlying statistical models.

Suppose two robotic commands

"*can you bring the book on the table*"

and

"*can you bring the volume on the table*"

that are supposed to express the same meaning. They differ each other just for the surface forms of the *book* concept. In fact, although according to WordNet [116] *book* and *volume* may evoke 15 and 6 different concepts respectively (or, in WordNet, *synsets*), they share at least one of them. That is, given the contexts in which they appear, one might exclude out of context meanings such as the action of reserving something (*to book*) or the magnitude of sound (*volume*). Hence, without a proper generalization of words representation, the sparse nature of the natural language lexicon does not allow to catch and compare the actual meaning of different surface forms.

In order to define a learning algorithm providing an effective lexical generalization without a strong dependency from hand-built resources, an automatic approach to acquire and generalize lexical information directly from data is here discussed. Such acquisition is managed through the distributional analysis of large scale corpora.

Linguistic phenomena, here words, are modeled according to a geometrical perspective, i.e., points in a high-dimensional space representing semantic concepts, in such a way that similar, or related, concepts are near each another in the space.

Distributional approaches represent lexical semantics through the analysis of observations in large-scale corpora. The fundamental intuition is that the meaning of a word can be described by the set of textual contexts in which it appears. It is commonly known as *Distributional Hypothesis* [75] and can be synthesized from the following statement in [148]:

Words with similar meanings will occur with similar neighbors if enough text material is available.

The idea is thus to acquire a synthetic representation of a targeted word w, considering all other words co-occurring with w, such that words sharing the same co-occurrences will be represented similarly. A lexical similarity function can be thus defined in terms of similarity between these representations. It is worth noting that a good approximation of the words distributional information can be achieved whenever a sufficient amount of observations is gathered.

The Word Space Model

In this thesis, the distributional representation of words is acquired according to a geometrical perspective, i.e., words are represented as vectors whose components reflect the corresponding contexts. This allows to define a high-dimensional space known as *Word Space*, where the distance among instances, i.e., words, reflects the lexical similarity, as described in [147]:

Vector similarity is the only information present in Word Space: semantically related words are close, unrelated words are distant.

Words are points in this space and whenever two words have similar contexts, they will have a similar representations and they will be close in the space.

From a linguistic perspective, they are likely to be related by some type of generic **semantic relation**, either *paradigmatic* (e.g., synonymy, hyperonymy, antonymy) or *syntagmatic* (e.g., meronymy, conceptual and phrasal association), as observed in [143].

From a computational perspective, a matrix M is acquired through the analysis of large corpora, with the rows describing words as vectors $\vec{w}_i$, and columns representing the corpus contexts $\vec{c}_j$. Hence, each entry $w_{i,j}$ will be a measure associating words to contexts. Given two words w_1 and w_2, a function accounting for their similarity can be estimated by evaluating the *Cosine Similarity* between the corresponding projections $\vec{w}_1$, $\vec{w}_2$:

$$\cos(w_1, w_2) = \frac{\vec{w}_1 \cdot \vec{w}_2}{||\vec{w}_1||||\vec{w}_2||} \tag{A.46}$$

thus measuring the angle between such vectors.

When building such representations, three different typology of context can be exploited. In the *Topical* space two words are expected to have a similar geometric representation if they tend to appear in the same documents of a corpus. The *Syntax-based* space aims at capturing paradigmatic relations by imposing strict

syntactic constraints over the context selection. Finally, the *Word-based* space aims at providing a distributional lexical model while capturing paradigmatic relations between *target words* (*tw*s). Paradigmatic relations concern substitution and relate entities that do not co-occur in the text; a paradigm is thus a set of such substitutable entities. In a Word-based space, vectors represent *tw*s, while dimensions are words appearing in a k-windows around the *tw*s [143]. Consider, for example, the adjectives *beautiful*, *attractive* and *pretty*. They are *synonyms* in phrases like *"the beautiful girl"*, *"the attractive girl"* or *"the pretty girl"*. This trivial example is already enough to catch the informative load brought by the context. In fact, it is straightforward noticing that these words co-occur with the word *"girl"*. Hence, whenever words can be exchanged in the accounted language without altering the meaning of a sentence, in a large-scale document collection they will tend to co-occur with in the same contexts. Correspondingly, if vector dimensions correspond to words in the corpus, in a Word-based space *tw*s co-occurring with the same set of words are similarly represented, having initialized almost the same set of geometrical components. However, this property does not hold just for synonyms, as words involved in a paradigmatic relation will benefit of the same properties. An example is provided by the terms *knife* or *rifle* that, though not synonyms, they can be exchanged in a text, as sharing a consistent subset of co-occurring words.

In this *words-by-words* matrix each item is a value counting the co-occurrences between a *tw* and other words in the corpus, within a given window of word tokens. The window width k is a parameter allowing the space to capture different lexical properties: larger values for k tend to introduce more words, i.e., possibly noisy information, whereas lower values lead to stricter forms of similarity/equivalence. Moreover, in order to capture shallow syntactic information, words co-occurring on the left context are treated separately from words occurring on the right one.

Lower Dimensional Vector Spaces

The quality of a Word Space is tied to the amount of information analyzed and the more contextual information is provided, the more accurate will be the resulting lexical representation. However, some problems of scalability arise when the number of the space dimension increases. From a computationally perspective, a space with thousand dimensions make the similarity estimation between vector expensive. Consequently, even a simple operation (e.g., the search of the most similar words to a target word) can be prohibitive in terms of computational cost. Moreover, from a geometric perspective, the notion of similarity between vectors is sparsely distributed in high-dimensional space.

Fortunately, employing a geometrical representation for words enables the adoption of dimensionality reduction techniques to reduce the complexity of the high-dimensional space. The resulting representation of the initial vector space will contain the same, but denser, information. In fact, the new dimensions can not be mapped directly to dimensions of the initial matrix; conversely, each dimension will represent a newly synthesized (and more informative) feature. Such techniques allow to exploit data, i.e., words and contexts, distribution and topology in order to acquire a more compact representation and define more meaningful data-driven metrics.

In the following, two approaches for lowering matrix dimensionality are presented: while the former is based on linear algebra factorization, the latter exploits non-linear combinations of weights in a neural net.

Latent Semantic Analysis An example of linear dimensionality reduction technique is Latent Semantic Analysis (LSA) [101]. The original *word-by-context* matrix M is decomposed through Singual Value Decomposition (SVD) [69] into the product of three new matrices: U, S, and V so that S is diagonal and

$$M = USV^T \tag{A.47}$$

M is then approximated to

$$M_k = U_k S_k V_k^T \tag{A.48}$$

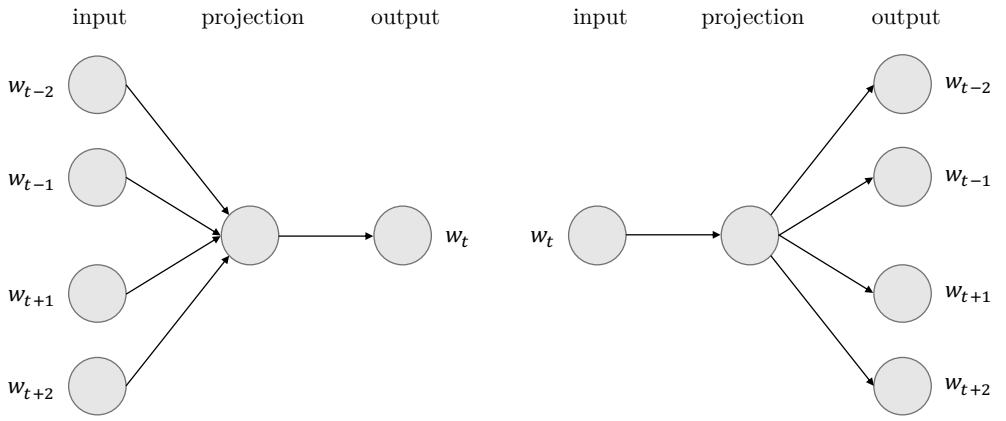
in which only the first k columns of U and V are used, so that only the first k greatest singular values are considered. This approximation supplies a way to project a generic term w_i into the k-dimensional space using $W = U_k S_k^{\frac{1}{2}}$, where each row $\vec{w}_i^k$ corresponds to the representation vectors $\vec{w}_i$. The original statistical information about M is captured by the new k-dimensional space which preserves the global structure while removing low-variant dimensions, i.e., distribution noise.

The lexical similarity can still be computed in such reduced space through the cosine similarity (Equation A.46), in a space with a reduced number of dimensions (e.g., $k = 250$) where the notion of distance is significantly more informative with respect to the original space. These newly derived features may be considered latent concepts, each one representing an emerging meaning component as a linear combination of many different original contexts.

Word Embeddings through Neural Networks The term word embeddings appeared for the first time in 2003, when Bengio et al. [22] proposed a new vector space trained through a neural language model. Later on, Collobert and Weston [37] demonstrated the power of pre-trained word embeddings, establishing word embeddings as a highly effective tool when applied to downstream tasks and proposing a neural network architecture that many of today's approaches are built upon. However, in 2013 Mikolov et al. [115] brought word embeddings to the forefront by designing, implementing and releasing a toolkit enabling the training and use of pre-trained embeddings, namely, *word2vec*.

word2vec implements the most popular toolkit for generating high-quality word vectors from huge datasets. In fact, it relies on the probabilistic feed-forward Neural Network Language Model (NNLM) proposed in [22] and defines two new different architectures A.10.

The first architecture A.10(a) is similar to the feed-forward NNLM. It aims at predicting the current word based on the context. While the non-linear hidden layer is removed, the projection layer is shared for all words. Hence, words are projected into the same position. This architecture is called a bag-of-words model as the order of words in the history does not influence the projection. As also words from the future are available, both previous and next words of the target one are considered as input. This further extension takes the name of Continuous Bag-Of-Word (CBOW), denoting that the model uses a continuous distributed representation of the context.



(a) CBOW (b) Skip-gram

Figure A.10. Model architectures proposed in *word2vec*

The second architecture A.10(b), called *Continuous Skip-gram model*, is similar to the previous one. However, differently from the CBOW, it tries to maximize the classification of a word based on another word in the same sentence. Operationally, given a word as input to a log-linear classifier, it predicts words of the previous and next context of a certain window. The range is selected as a trade-off between word vectors quality and computational complexity.

A.2. Machine Learning for Visual Perception

As we stated before, HRI is a research area where many different interaction modalities are being studied. Understanding what we say, following our to make a guess about what we are talking about, catching our finger pointing an object to disambiguate persisting ambiguities are essential capabilities that a robot should provide, in order to intelligently interact with users. In fact, interactions require the perception of a signal, be it audio, visual, or haptic.

This section focuses on vision processing, again, an essential component in any modern robotic platform. For instance, vision capabilities allow the robot to actively perceive the environment, as well as to identify and recognize the entities therein. Hence, this capability may play a key role in creating structured representations of the environment, supporting the Human Augmented Semantic Mapping. A major requirement enabling this feature is the access to a sensing system (e.g., RGB-D vs. stereo cameras, laser scanners, ...), even though most of the commercial/research robotic platforms are already equipped with this kind of devices, as well as the software layer devoted to processing such input signals.

Recently, a huge effort is being spent on the research on object recognition for robotic applications. The reason for such a growing interest is due to several elements. First, object detection and recognition is a capability that has a wide applicability in real scenarios, ranging from mobile applications up to the robotic domain. Moreover, the continuously increasing availability of large scale datasets helped in promoting the application of deep learning techniques to this problem.

At the same pace, the development of new ad-hoc architectures of neural networks provided a huge boost.

For example, in [153, 182], deep learning techniques are exploited to detect and recognize objects in a real scene. The architecture dominating image classification is the Convolutional Neural Network (CNN), that seems to outperform any other architecture thanks to its convolutional layer. Other approaches are based on transfer learning techniques, as the zero-shot learning [83, 100] or one-shot learning [91]. However, these approaches are focused on attribute-learning; as a consequence, a pre-trained visual classifier for extracting attributes is required. Some works proposed to apply kernel-based learning algorithms to object classification. For example, in [99, 129, 149], SVMs are trained to recognize objects in real scenarios. In particular, a combined CNN/SVM approach is proposed in [149], where a pre-trained CNN is used to automatically generate feature vectors, whereas a SVM perform the actual classification of the scene. However, the works that are closer to the purpose of this thesis are the ones leveraging Human-Robot Interaction (HRI) to acquire objects label [91, 131, 151]. The main idea is that once a new item is shown, the robot starts a dialogic interaction with the tutor to acquire the corresponding label. Hence, the dataset is provided incrementally, and the more interaction, the bigger will be the training set.

Though promising, most of the cited approaches work on fully labeled training dataset, and the number of classes, along with their categories, must be known in advance. Hence, the main goal is thus to achieve good performance on a set of predefined objects, rather than focusing on learning new categories once the system is in operation. This represents a major limitation for a real scenario application, as a complete prior knowledge about what one may find within an environment is a constraint rather unrealistic.

To overcome such limitation, a class of incremental neural networks is introduced in the following, enabling the incremental acquisition of unseen objects' categories.

A.2.1. Load-Balancing Self-Organizing Incremental Neural Network

Load-Balancing Self-Organizing Incremental Neural Network (LB-SOINN) [188] is an unsupervised learning algorithm based on Self-Organizing Incremental Neural Network (SOINN) [58] that is, in turn, built upon the concept of Self-Organizing Map.

This method allows to learn a model that reflects the underlying topology of the data distribution, without the need to provide number and label of classes in advance. In fact, being an unsupervised classification method (data clustering), the problem is defined as finding homogeneous groups of data points in a given multidimensional data set.

The underlying idea is that each node in the network has an associated weight which lives in the feature space. Every time a new image is input to the vision module, the LB-SOINN algorithm assesses whether a new node has to be added to the network, based on the feature vector similarity to all the other nodes' associated weights. If no node is added, then the closest node and its neighbors' weights are updated, and the two closest nodes are joined by an edge. In this manner, the

structure of the network evolves to reflect how the data is distributed in the feature space.

LB-SOINN is a further improvement of the Enhanced Self-Organizing Incremental Neural Network (E-SOINN) [59], whose main limitation are (i) a strong dependency on the sequence of input data, (ii) instability of the learning algorithm, causing multiple (and not required) combination and separation of high-density overlapped areas and (iii) the Euclidean distance is used as metric to find the nearest node. These limitations are overcome in LB-SOINN, by introducing the following novelties: (i) *load balancing between nodes*, to alleviate dependency on the sequence of input data; (ii) *combination and separation of subclasses based on Voronoi tessellation*, to avoid multiple combinations and separation of subclasses; (iii) a combination of different distance metrics (i.e., Euclidean distance and cosine similarity), to defeat the curse of dimensionality.

For further details on LB-SOINN, the reader should refer to [188].