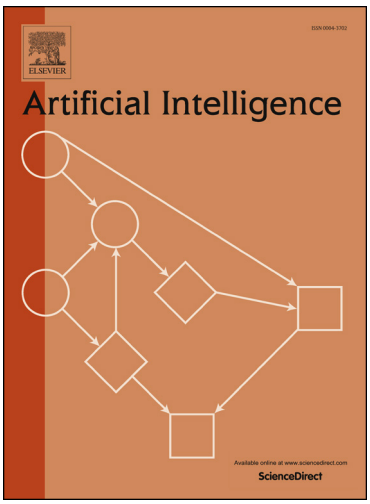
Grounded Language Interpretation of Robotic Commands through Structured Learning

Journal Item

Journal Pre-proof



Grounded Language Interpretation of Robotic Commands through Structured Learning

Andrea Vanzo, Danilo Croce, Emanuele Bastianelli, Roberto Basili, Daniele Nardi

PII: S0004-3702(18)30293-5
DOI: https://doi.org/10.1016/j.artint.2019.103181
Reference: ARTINT 103181

To appear in: *Artificial Intelligence*

Received date: 6 June 2018
Revised date: 17 September 2019
Accepted date: 1 October 2019

Please cite this article as: A. Vanzo et al., Grounded Language Interpretation of Robotic Commands through Structured Learning, *Artif. Intell.* (2019), 103181, doi: https://doi.org/10.1016/j.artint.2019.103181.

Grounded Language Interpretation of Robotic Commands through Structured Learning

Andrea Vanzo[a,*], Danilo Croce[b], Emanuele Bastianelli[c,], Roberto Basili[b], Daniele Nardi[a]

[a] *Dept. of Computer, Control and Management Engineering "Antonio Ruberti", Sapienza University of Rome, Italy*
[b] *Dept. of Enterprise Engineering, University of Rome "Tor Vergata", Italy*
[c] *Knowledge Media Institute, The Open University, Milton Keynes, UK*

Abstract

The presence of robots in everyday life is increasing day by day at a growing pace. Industrial and working environments, health-care assistance in public or domestic areas can benefit from robots' services to accomplish manifold tasks that are difficult and annoying for humans. In such scenarios, Natural Language interactions, enabling collaboration and robot control, are meant to be situated, in the sense that both the user and the robot access and make reference to the environment. Contextual knowledge may thus play a key role in solving inherent ambiguities of grounded language as, for example, the prepositional phrase attachment.

In this work, we present a linguistic pipeline for semantic processing of robotic commands, that combines discriminative structured learning, distributional semantics and contextual evidence extracted from the working environment. The final goal is to make the interpretation process of linguistic exchanges depending on physical, cognitive and language-dependent aspects. We present, formalize and discuss an adaptive Spoken Language Understanding chain for robotic commands, that explicitly depends on the

*Corresponding author now at Heriot-Watt University, Edinburgh, Scotland, UK.
**Work partially done while the author was at Dept. of Civil Engineering and Computer Science Engineering, University of Rome "Tor Vergata", Italy
Email addresses: vanzo@diag.uniroma1.it (Andrea Vanzo), croce@info.uniroma2.it (Danilo Croce), emanuele.bastianelli@open.ac.uk (Emanuele Bastianelli), basili@info.uniroma2.it (Roberto Basili), nardi@diag.uniroma1.it (Daniele Nardi)

operational context during both the learning and processing stages. The resulting framework allows to model heterogeneous information concerning the environment (e.g., positional information about the objects and their properties) and to inject it in the learning process. Empirical results demonstrate a significant contribution of such additional dimensions, achieving up to a 25% of relative error reduction with respect to a pipeline that only exploits linguistic evidence.

Keywords: Spoken Language Understanding, Automatic Interpretation of Robotic Commands, Grounded Language Learning, Human-Robot Interaction

1. Introduction

In the last decade, Human-Robot Interaction (HRI) is getting more and more attention within the AI and Robotics community. In fact, several different motivations are pushing forward the breakthroughs in the field. First, HRI embraces an incredibly wide range of research interests and topics. A domestic robot is expected of being able to: (i) navigate and self-localize within the environment, (ii) recognize people and objects (Vision capabilities), (iii) manipulate physical items (Grasping) and (iv) properly interact with human beings (Human-Robot Interaction). All these different challenges involve several capabilities (and so paradigms) that need to coherently interplay in order to design and build proper interactive robots. Second, domestic robots are going to be part of our everyday life in the very next future. Several robotic platforms have been already marketed and, at different level of specificity, they are able to support a variety of activities. The *iRobot Roomba* is probably the best among possible examples, due to its commercial success and the amount of innovation it contributed with. It is a vacuum cleaner capable of building a map of the environment, in order to autonomously plan and execute the cleaning of our homes.

However, though such a way of interacting with the robotic platform might be considered direct and accessible, human language is still one of the most natural ways of communication for its expressiveness and flexibility: the ability of a robot to correctly interpret users' commands is essential for proper HRI. For example, a spoken language interface would make the *Roomba* accessible to even more users.

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" (1)

a number of implicit assumptions should be met in order to enable the robot to successfully execute the command. First, the user refers to entities that must exists 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.

We argue 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 1 is ambiguous with respect to both syntax and semantics due to the Prepositional Phrase (PP) attachment ambiguity ([1, 2]). In the running example 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]]] (2)

[VP take [NP the mug] [PP next to the keyboard]] (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 [3], whenever the syntactic structure of the running command is 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 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.

In fact, the structured representation of the environment is a discriminating factor for resolving syntactic/semantic ambiguities of language, such as the attachment of the PP *"next to the mug"*, as well as for providing the

required knowledge in support of language grounding in a situated scenario. While such ambiguities can be resolved through interactions, we believe that, when useful resources are available, a knowledgeable system should exploit them in order to minimise the user annoyance.

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

1.1. Contributions and article outline

The main contribution of this article consists of a framework for the automatic understanding of robotic commands, aimed at producing interpretations that coherently mediate among the world, the robotic platform and the pure linguistic level triggered by a sentence. In fact, we support the idea that the interpretation of a robotic command is not just an outcome of a linguistic inference, but it is the result of a joint reasoning process involving both linguistic evidence and knowledge regarding the contextual physical scenario. This work builds upon [4], that shows how the interpretation process of a command can be made sensitive to the spatial position of perceived entities within the environment. Here we make a step forward by proving that the interpretation framework can be extended to richer feature spaces, that allow for expressing domain properties of the involved entities, along with spatial ones. To this end, this paper provides a robust formalization of the Semantic Map, that collects all the semantic properties to be injected in the language understanding process. Moreover, we prove the approach to be language independent, with a more complete experimental session run over a corpus in two different languages (i.e., English and Italian). Hence, the proposed approach allows to (i) learn the interpretation function by relying on a corpus of annotated commands, (ii) inject grounded information directly within the learning algorithm, thus integrating linguistic and contextual knowledge, and (iii) extend the features space as more specific and rich information is made available. Experimental evaluations show that the injection of these dimensions in the interpretation process is beneficial for the correct interpretation of the real user intent, when perceptual knowledge is paired with information coming from the operational domain.

We organize the manuscript in 7 sections. In the next section, the problem of natural language interpretation grounded in a robotic operating en-

vironment is discussed in the view of previous research and achievements in literature. Section 3 provides a description of the knowledge resources, referring to both the linguistic assumptions and context modeling, while Section 4 describes the grounding process we designed, which allows to link linguistic symbols to entities into the environment. In Section 5 a formal description of the proposed system is provided, together with the adopted machine learning techniques to feature modeling; results obtained through several experimental evaluations are reported in Section 6. Finally, in Section 7 we draw some conclusions.

2. Related Work

The approach we propose makes use of grounded features extracted from a Semantic Map [5] modeling the entities in the environment, as well as *semantic* and *spatial* properties. Such features allow to drive 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 [6, 7]).

Grounding language often requires the combination of the linguistic dimension and perception. For example, in [8], the authors make a joint use of linguistic and perceptual information. Their approach leverages active perception, so that linguistic symbols are directly grounded to elements actively perceived. Again, in [9], a Natural Language Understanding system called Lucia is presented, based on Embodied Construction Grammar (ECG) 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 these works perceptual knowledge never modifies syntactic structures that can be generated by the parser when they are incorrect. Conversely, our system is able to deal with ambiguities at predicate level, allowing for selecting the interpretation that is mostly coherent with the operational environment.

Similarly to our framework, the approaches in [10, 11] aim at grounding language to perception through structured robot world knowledge. In particular, in [11] the authors deal with the problem of using unknown out-

of-vocabulary words to refer to objects within the environment; the meaning of such words is then acquired through dialog. Differently, we make use of a mechanism based on Distributional Model of Lexical Semantics [12, 13] together with phonetic similarity functions to achieve robustness (as in [14]), while extracting grounded features through the lexical references contained in the Semantic Map. Thanks to this mechanism, no further interactions are required, and the acquisition of synonymic expressions 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 [15]. 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 [16], 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 [17] 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 [18], 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 large use of perceptual information to solve an 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 [19] the authors face the problem of teaching a robot manipulator how to execute natural language commands by demonstration, using video/caption pairs as valuable source of information. Our system relies on a synthetic representation of the environment, acquired through active interaction [20]. It allows the robot to make inferences on the world it is working into, though it is not actively and directly observing the surrounding environment. However, since the perception is injected in the interpretation process as features for the learning machine, the framework we propose can be scaled to active perception, whenever vision information can be extracted and encoded into features in real-time.

A different perspective has been addressed in [21], where the problem of PP attachment ambiguity 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 [22] 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, we use a synthetic representation of the context to resolve semantic ambiguities of the human language, with respect to a situated interactive scenario. Our approach is thus able to cope with the correct semantics of a 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 [23, 24, 25]. Although the underlying idea of these works is similar to ours, our aim is to produce an interpretation at the predicate level, that can in turn be grounded in a robotic plan corresponding to the action expressed in an utterance. Therefore, the findings of such works can be considered as complementary to our proposal, as while they focus just on grounding linguistic symbols into entities and attributes, we leverage such a process for linking the whole interpretation to the current world.

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

- The perceptual information we leverage 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 restructuring the overall system. Moreover, such property is useful to evaluate the contributions provided by individual features.

- In our 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.

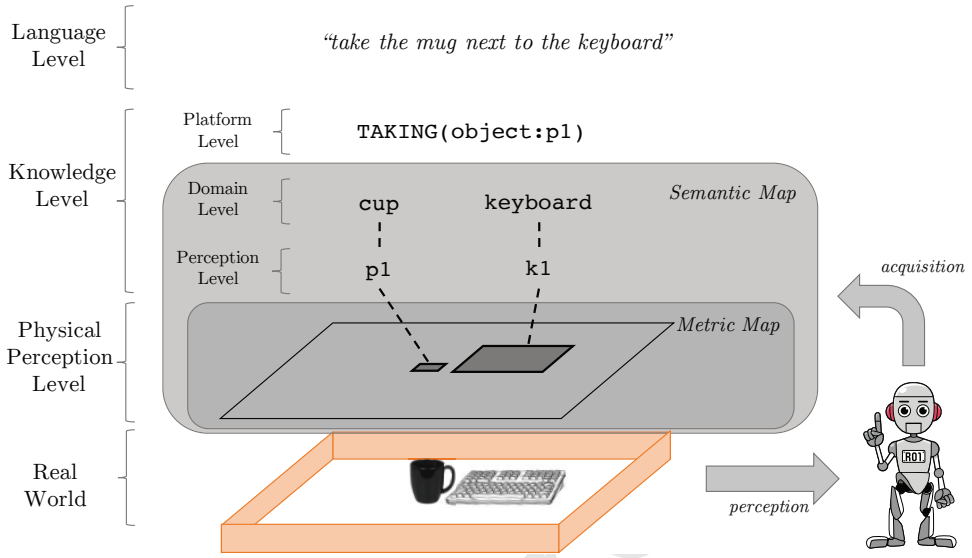


Figure 1: Layered representation of the knowledge involved in the interpretation of robotic commands

- 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 Distributional Model of Lexical Semantics, 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.

3. Knowledge, Language and Learning 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 in HRI depends on a variety of other factors, including the perception of the environment. We categorize these factors into a layered representation as shown in Figure 1. First, we consider the *Language Level* as 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*: we assume that the robot has 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 this information plays a crucial role during linguistic interactions. The integration of metric information with notions from the knowledge level provides an augmented representation of the environment, called *Semantic Map* [5]. 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) [26, 20]. 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, we introduce an additional level (namely *Platform Level*), whose information is instantiated in a knowledge base called *Platform Model* (PM). The main aim of such a knowledge base is to enumerate all the actions the robot is able to execute. While SLU for HRI has been mostly carried out over evidence specific to the linguistic level, e.g., in [27, 28, 29, 30], this process should deal with all the aforementioned layers in a 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, we introduce the building blocks of our perceptually informed framework, defining the adopted interpretation formalism and shaping the perceptual information in a structured representation, i.e., the Semantic Map.

3.1. Frame-based Interpretation

A command interpretation system for a robotic platform must produce interpretations of user utterances. As in [31], the understanding process is

based on the Frame Semantics theory [32], which allows usto give a linguistic and cognitive basis to the interpretations. In particular, we consider the formalization promoted in the FrameNet [33] project, 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.

Let us consider the running example 1 *"take the mug next to the keyboard"* provided in Section 1. 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

$$[take]_{\textbf{\textit{Taking}}}\ [the\ mug\ next\ to\ the\ keyboard]_{\textsc{Theme}} \tag{4}$$

or

$$[take]_{\textbf{\textit{Bringing}}}\ [the\ mug]_{\textsc{Theme}}\ [next\ to\ the\ keyboard]_{\textsc{Goal}} \tag{5}$$

depending on the configuration of the environment.

In the following, we introduce the notation used for defining an interpretation in terms of semantic frames and that 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., $s = (w_1, ..., w_{|s|})$, an interpretation

$\mathcal{I}(s)$ in terms of semantic frames determines a conjunction of predicates as follows:

$$\mathcal{I}(s) = \bigwedge_{i=1}^{n} p^i \tag{6}$$

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

$$p^i = \langle f^i, Arg^i \rangle \tag{7}$$

where:

- $f^i \in F$ is the frame of the ith predicate evoked by the sentence, where F is the set of possible frames as defined in the Platform Model, 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 4, while $[$*the mug*$]_{\text{THEME}}$ and $[$*next to the keyboard*$]_{\text{GOAL}}$ for the interpretation 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: $a^i_j = (w_m, \ldots, w_n)$ with $1 \leq m < n \leq |s|$, e.g., *"the mug next to the keyboard"* for 4 or *"the mug"* and *"next to the keyboard"* for 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 4 and 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 4 or *"mug"* and *"keyboard"* for the arguments of interpretation 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 1 will be represented through the following structures

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

or

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

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.

3.2. Semantic Map

In this section we describe how to properly represent the environmental knowledge required for the interpretation process and provided by the robot. In line with [34] and according to the layered representation provided at the beginning of Section 3, we structure the Semantic Map (Figure 1) as the triple:

$$\mathcal{SM} = \langle \mathcal{R}, \mathcal{M}, \mathcal{P} \rangle \tag{8}$$

such as:

- $\mathcal{R}$ 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 $\mathcal{M}$.

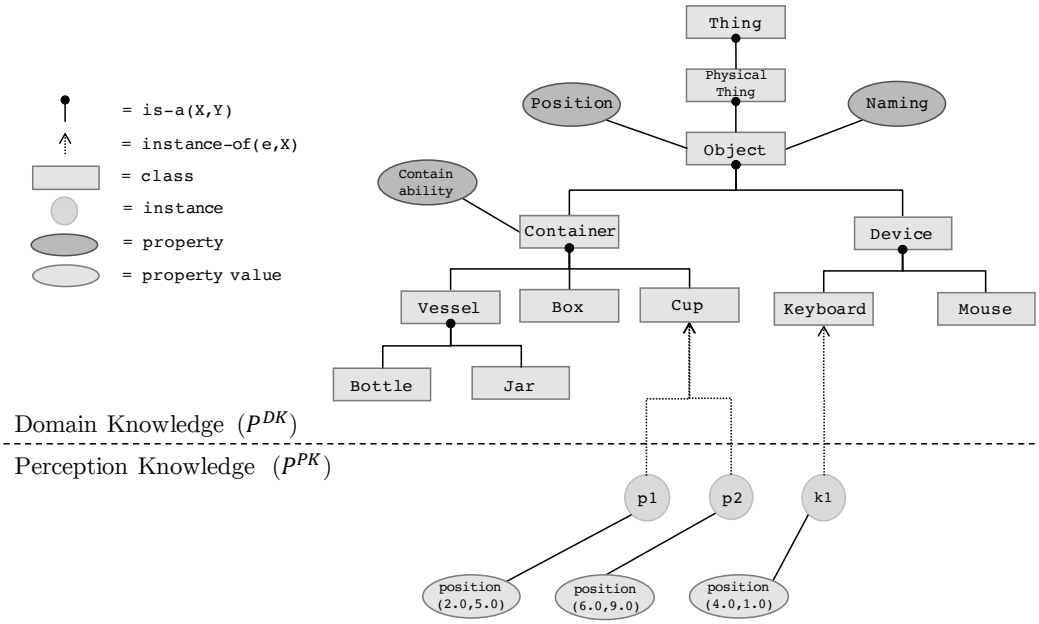


Figure 2: Sketch of the knowledge contained into a Semantic Map

$\mathcal{P}$ is modeled as a *(Monotonic) Inheritance Network*. It is worth emphasizing that we do not require that the knowledge acquired through perception is fully consistent with the taxonomy of classes, as the Semantic Map is only used to support the linguistic processes addressed in this article. Hence, we decompose $\mathcal{P}$ into two layers:

$$\mathcal{P} = \langle \mathcal{P}^{\mathcal{DK}}, \mathcal{P}^{\mathcal{PK}} \rangle \tag{9}$$

where:

- $\mathcal{P}^{\mathcal{DK}}$ is the *Domain Knowledge*, 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;
- $\mathcal{P}^{\mathcal{PK}}$ is the *Perception Knowledge*, collecting entities and properties specific of the targeted environment and representing the extensional knowledge, acquired by the robot.

The resulting structure of $\mathcal{P}$ is shown in Figure 2, highlighting both the Domain Knowledge $\mathcal{P}^{\mathcal{DK}}$ and the Perception Knowledge $\mathcal{P}^{\mathcal{PK}}$.

The Semantic Map generation can follow different approaches: by relying on hand-crafted ontologies and using traditional AI reasoning techniques [35, 36], by exploiting the purely automatic interpretation of perceptual outcomes [37, 38, 39], or by relying on interactions in a human-robot collaboration setting [40, 41]. However, the creation of the Semantic Map is out of the scope of this paper and we assume it as an available resource of the robotic system providing gold information. In fact, it is worth noting that the Semantic Map is an essential component of any real robot. Active perception mechanisms such as Computer Vision systems based on Deep Learning still lack in providing robust understanding of the surrounding world to support reasoning and planning mechanisms.

Domain Knowledge. The *Domain Knowledge* provides the terminology of the Semantic Map. It allows to define and structure the knowledge shared by different environments in the same domain. Such a resource can be either automatically generated consulting existing resources (e.g. WordNet [42] or ConceptNet [43]), extracted from unstructured documents (e.g. from texts present on the Web [44]), or manually created by a knowledge engineer.

In particular, the Domain Knowledge proposed here (Figure 2, upper part) is built upon the WordNet taxonomy and aims at modeling the hierarchy of classes related to a domestic environment, and the domain-dependent semantic attributes.[1]

To model the Domain Knowledge $\mathcal{P}^{\mathcal{DK}}$, we use `is-a` to define the hierarchy of classes, e.g., `is-a`(`Cup`, `Container`), and three specific properties: *Contain-ability*, *Naming* and *Position*. *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 included into the Domain Knowledge:

- `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*}).

[1] We assume the attributes to be part of the Domain Knowledge, as active perception of those features is out of the scope of the article.

For the *Contain-able* property, the *Closed World Assumption* is applied, 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 by 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 our setting, alternative naming has been provided by the combined analysis of Distributional Models and Lexical Databases (e.g., WordNet), and validated by a knowledge engineer.

Perception Knowledge. The Perception Knowledge $\mathcal{P}^{\mathcal{PK}}$ (Figure 2, 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 into the environment and perceived by the robot through its sensors.

$\mathcal{P}^{\mathcal{PK}}$ is defined through `instance-of(e, C)`, meaning that entity `e` is an entity of 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(`e1, e2`) allows to return the Euclidean distance among the entities `e1` and `e2`. This value is essential to determine whether two entities are far or near into the environment and possibly change the assumptions made during the interpretation of sentences making reference to these entities. For example, given two entities `entity-of(p1, Cup)` and `entity-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(`p1, k1`) $= 4.47$.

4. Grounding: a Side Effect of Linguistic Interpretation and Perception

When interacting with a robot, users make references to the environment. 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 $\mathcal{P}^{\mathcal{PK}}$, collected as:

$$\mathcal{E}_{\mathcal{PPK}} = \{\mathtt{e} \mid \mathtt{instance\text{-}of}(\mathtt{e}, \cdot)\} \tag{10}$$

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

$$\mathcal{N}(\mathtt{e}) = \{w_\mathtt{e} \mid \mathtt{instance\text{-}of}(\mathtt{e}, \mathtt{C}) \wedge \mathtt{naming}(\mathtt{C}, \mathtt{N}) \wedge w_\mathtt{e} \in \mathtt{N}\} \tag{11}$$

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

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 the cosine similarity 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:

$$\Gamma(arg^i_j, \mathcal{P}^{\mathcal{PK}}) \rightarrow \mathcal{G}_{arg^i_j} = \{\mathtt{e} \in \mathcal{E}_{\mathcal{PPK}} \mid \exists w_\mathtt{e} \in \mathcal{N}(\mathtt{e}) \wedge g(h, w_\mathtt{e}) > \tau\} \tag{12}$$

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

The lexical semantic vectors are acquired through corpus analysis, as in Distributional Lexical Semantic paradigms. 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 of the same class are present into the environment and they have different colors. 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.

It is worth noting that the grounding mechanism is here used to support the disambiguation of ambiguous commands, and it does not constitute the main contribution of the paper. Moreover, being such a process completely decoupled from the semantic parsing, different approaches for g (and therefore of Γ) can be designed by relying on just linguistic evidence [14] or visual features [45]. However, the proposed mechanism is extensively used in this article to locate candidate grounded entities in the Semantic Map and to code them into perceptual features in the understanding process, described below.

5. Perceptually Informed Interpretation: the Language Understanding Cascade

The interpretation framework we propose is based on a cascade of statistical classification processes, modeled as sequence labeling tasks (Figure 3). The classification is applied to the entire sentence and is modeled as the Markovian formulation of a structured SVM (i.e., SVM^{hmm} proposed in [46]). 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 $\mathbf{x} = (\vec{x}_1 \dots \vec{x}_l) \in \mathcal{X}$, where $\mathbf{x}$ is a sentence and $\vec{x}_i \in \mathbb{R}^n$ 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 Argument Identification and Argument Classification sub-steps. Each of these is mapped into a different SVM^{hmm} sequence labeling task.

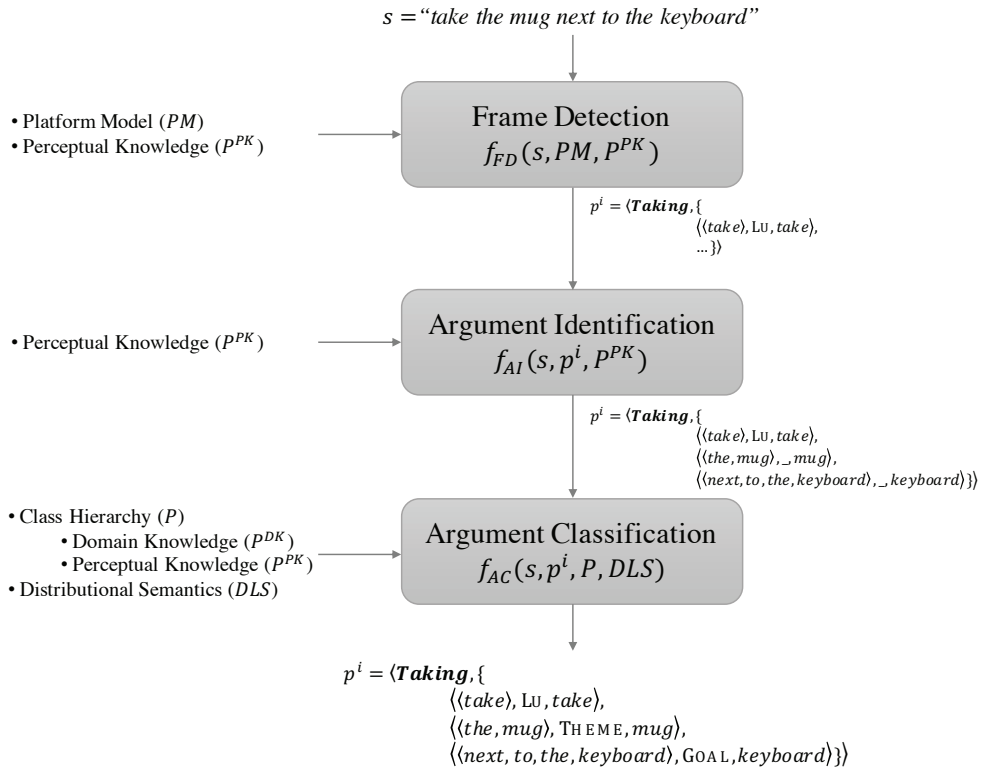


Figure 3: Processing cascade modeling the interpretation task

In the following, we first introduce the ML approach and then address its application to each step of the cascade.

5.1. 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., $x_{i-k}, \ldots, x_{i-1}$. Given this history, a sequence of k step-specific labels can be retrieved, in the form $y_{i-k}, \ldots, y_{i-1}$. In order to make the classification of x_i dependent also from the history, we augment the feature vector of x_i introducing a vector of transitions $\psi_{tr}(y_{i-k}, \ldots, y_{i-1}) \in \mathbb{R}^l$: 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:

$$x_i^k = \phi(x_i; y_{i-k}, \ldots, y_{i-1}) = \psi_{obs}(x_i) \;||\; \psi_{tr}(y_{i-k}, \ldots, y_{i-1}) \tag{13}$$

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 $x_{j_a}||x_{j_b}$:

$$K(x_i, x_j) = K_a(x_{i_a}, x_{j_a}) + K_b(x_{i_b}, x_{j_b}) = \langle x_{i_a}||x_{i_b}, x_{j_a}||x_{j_b}\rangle \tag{14}$$

Conversely, $\psi_{obs}(x_i) = x_{i_a}||x_{i_b}$.[2]

At training time, we use the SVM learning algorithm LibLinear, proposed in [47] and implemented in KeLP [48] 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:

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

[2]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.

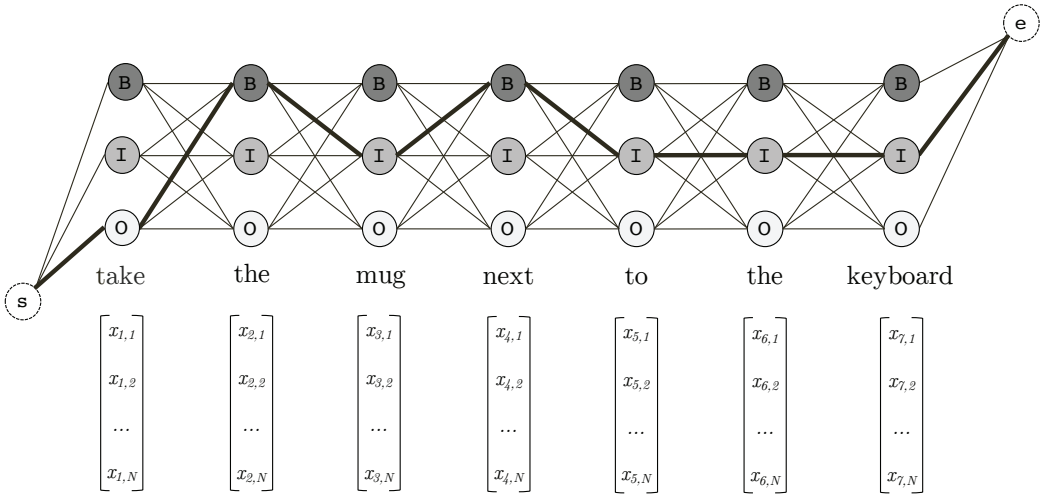


Figure 4: Viterbi decoding trellis of the Argument Identification step (Section 5.3), for the running command "take the mug next to the keyboard", when the interpretation 5 is evoked. The label set refers to the IOB2 scheme, so that $y_i \in \{B, I, O\}$. Feature vectors x_i are obtained through the 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.

In order to reduce the computational cost, a *Viterbi-like decoding algorithm* (Figure 4) is adopted[3] to derive the sequence, and thus build the augmented feature vectors through the ϕ function.

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

5.2. Frame Detection

Our 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, PM is the Platform Model and $\mathcal{P}^{\mathcal{PK}}$is the Perception Knowledge. Assuming $s =$"*take the mug*

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

next to the keyboard", then

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

for interpretation 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.

As already explained, the labeling process depends on linguistic information, as well as on the information derived from the Platform Model (i.e., actions the robot is able to execute) and perceptual features extracted from the $\mathcal{P}^{\mathcal{PK}}$. In our 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 hereafter detailed.

5.2.1. 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 POS-tag of the current word or the contextual POS-tag n-grams).

5.2.2. Robot-dependent features

Information about the robot coming from the PM are used to represent executable actions: these are mapped into frames through their corresponding LUs. The 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 PM), then, for every frame $f^i \in F^k$, the corresponding i-th feature of the w_k is set to `true`.

5.2.3. Perceptual features

In addition, features derived from the perceptual knowledge 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, $n(v_h) = \{w_k \mid POS(v_h) == \text{VB} \wedge POS(w_k) == \text{NN} \wedge w_k$ is rooted in v_h in the dependency (sub)tree$\}$. Let $t(v_h)$ be the set of tokens governed by the verb v_h, $t(v_h) = \{t_k \mid POS(v_h) == \text{VB} \wedge t_k$ is rooted in v_h 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 a piece of perceptual evidence, its contribution must be considered when paired with another feature, whose aim is to explicit the number of entities that have been grounded by the tokens $w_k \in n(v_h)$. This feature is again 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 $|\bigcup_{\forall w_k \in arg_j^i \wedge w_k \in n(v_h)} \mathcal{G}_{arg_j^i}|$.

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.

5.3. Argument Identification

For each identified predicate $p^i \in \mathcal{I}(s)$, the **Argument Identification** (AI) 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 $arg^i_j \in Arg^i$. More formally, for a given sentence s, the i^{th} identified predicate p^i, the AI process can be summarized as the function $f_{AI}(s, p^i, \mathcal{P}^{\mathcal{PK}})$ updating the structure of $\mathcal{I}(s)$ as follows:

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

for interpretation 4, or

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

for interpretation 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 AI 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. 4)

or

`O`-*take* `B`-*the* `I`-*mug* `B`-*next* `I`-*to* `I`-*the* `I`-*keyboard* (Interpr. 5)

5.3.1. 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 (see Section 5.2.1).

5.3.2. Perceptual features

Similarly to the FD step, the following dedicated features derived from the perceptual knowledge are introduced.

Grounded entities. For each noun $w_k \in arg_j^i$ such that $\mathcal{G}_{arg_j^i} \neq \emptyset$, a boolean feature is set to `true`. It is worth reminding that $\mathcal{G}_{arg_j^i}$ contains candidate entities referred by arg_j^i. Moreover, for each preposition $prep_k$, given their syntactic dependent $w_k^{dep} \in arg_j^i$, a boolean feature is set to `true` if and only if $\mathcal{G}_{arg_j^i} \neq \emptyset$. Again, for each preposition $prep_k$, the number of nouns $w_k \in arg_j^i$ on the left and on the right of $prep_k$, whose $\mathcal{G}_{arg_j^i} \neq \emptyset$, 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 $w_f^{gov} \in arg_j^i$ and measure the average Euclidean distance in $\mathcal{P}^{\mathcal{PK}}$ between entities in $\mathcal{G}^{dep} \cup \mathcal{G}^{gov}$. As 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`.

5.4. 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 $arg_j^i \in Arg^i$ are labeled according to their semantic role $r_j^i \in arg_j^i$, 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. The main novelty of this work with respect to [4] 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 that might possibly contribute to the task achievement. Consequently, AC predictions will reflect also information extracted from the Domain Knowledge.

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

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

for interpretation 4, or

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

for interpretation 5.

5.4.1. Linguistic features

Again, the same morpho-syntactic features adopted in both FD and AI 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 (see Section 5.2.1).

In addition, Distributional Lexical Semantics (DLS) is applied to generalize the argument semantic head h_j^i of each argument arg_j^i: the distributional (vector) representation for h_j^i is thus introduced to extend the feature vector corresponding to each $w_k \in a_j^i$, where a_j^i is a member of the triple $\langle a_j^i, r_j^i, h_j^i \rangle = arg_j^i \in Arg^i$, representing the argument span.

5.4.2. 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 the Domain knowledge. 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 [4].

Entity-type attribute. The *Entity-type attribute* helps in generalizing the semantic head of an argument through the class the corresponding grounded entity belongs to. Again, for each p^i and for each $arg_j^i \in Arg^i$, the semantic head h_j^i is grounded into a set of possible candidate entities through $\mathcal{G}_{arg_j^i}$. The most promising candidate `e`, i.e., $\max_e g(h_j^i, w_e)$, is extracted and its class `C`, obtained through the predicate `is-a`$(\texttt{e}, \texttt{C})$, is applied to the semantic head feature vector.

Contain-ability attribute. The *Contain-ability attribute* 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 $arg_j^i \in Arg^i$, the semantic head h_j^i is grounded into a set of possible candidate entities through $\mathcal{G}_{arg_j^i}$. The most promising candidate `e`, i.e., $\max_e g(h_j^i, w_e)$, is then extracted and a boolean feature is applied to the semantic head feature vector, reflecting the value of `is-contain-able`$(\texttt{C}, \texttt{t})$, where `C` is the class the entity `e` belongs to.

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

Table 1: Feature modeling of the three steps (i.e., FD, AI and AC)

A reader-friendly sum up is provided in Table 1 where, for each step of the processing cascade, features and resources used are shown. In particular, while AI uses only *Linguistic features* and *Perception Knowledge* $\mathcal{P}^{\mathcal{PK}}$, in FD even the Platform Model PM is exploited. Conversely, due to the nature of the task the AC step mostly relies on Domain Knowledge $\mathcal{P}^{\mathcal{DK}}$ and Distributional Lexical Semantics DLS, in order to provide effective generalization capability while choosing the correct semantic role.

6. Experimental Evaluation

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 Appendix A.

The DLS vectors used in the grounding mechanism $g(\cdot, \cdot)$ have been acquired through a Skip-gram model [13], through the `word2vec` tool. By applying the settings *min-count=50*, *window=5*, *iter=10* and *negative=10* onto the UkWaC corpus [49], we derived 250 dimensional word vectors for more than $110,000$ words. The SVM^{hmm} algorithm has been implemented within the KeLP framework [48].

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, AI and AC, (Sections 6.1, 6.2 and 6.3). In these tests, we assume gold annotations as input information for the task, even if they depend on a previous processing step. The last test (Section 6.4) concerns the analysis of the 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 Perception Knowledge $\mathcal{P}^{\mathcal{PK}}$ is involved in both the FD and AI tasks, AC relies just on the Domain Knowledge $\mathcal{P}^{\mathcal{DK}}$and the Distributional Model DLS. Hence, in order to emphasize the contribution of such information, we considered two settings.

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 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 the Distributional Model DLS, 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 AI 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 1.

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

6.1. Frame Detection

In this experiment, we aim at 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.

		FD			
		Precision	**Recall**	**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 2: FD results: evaluating the whole span

Table 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 the 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.

However, when perceptual features (extracted from the Perception Knowledge $\mathcal{P}^{\mathcal{PK}}$) are injected, the F1 increases up to 95.94%, with a Relative Error Reduction 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 1, due to the *PP attachment* ambiguity, the interpretation of such verb may differ (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 in the Italian dataset.

6.2. Argument Identification

In this section, we evaluate the ability of the AI classifier in identifying the argument spans of the commands' predicates. According to the results reported in Table 3, this task seems to be the most challenging one. In fact,

		AI			
		Precision	**Recall**	**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 3: AI results: evaluating the whole span

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 the Perception Knowledge 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.

6.3. Argument Classification

For the scope of the article this experiment is the most interesting one, as here we inject the novel information extracted from the Domain Knowledge $\mathcal{P}^{\mathcal{DK}}$, regarding the Contain-ability property and the class of the grounded entity.

		AC			
		Precision	**Recall**	**F1**	**RER**
EN	*pLing*	$94.46\% \pm 0.05$	$94.46\% \pm 0.05$	$94.46\% \pm 0.05$	-
	Ground	$95.49\% \pm 0.05$	$95.49\% \pm 0.05$	$95.49\% \pm 0.05$	18.65%
IT	*pLing*	$91.52\% \pm 0.23$	$91.52\% \pm 0.23$	$91.52\% \pm 0.23$	-
	Ground	$92.21\% \pm 0.11$	$92.21\% \pm 0.11$	$92.21\% \pm 0.11$	8.14%

Table 4: AC results: evaluating the whole span

As reported in Table 4, the system is able to recognize the involved entities with high accuracy, with a 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.

Besides obtaining high 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, the *DLS* 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, given the training sentence "*take the book*", if *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, we provide further lexical generalization 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* 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"*.

6.4. End-to-End Processing Cascade

In this section, we conclude our 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.

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

Table 5: Evaluating the end-to-end chain against the whole span

In this configuration, we chose to report only the results of the AC step (Table 5), as its output represents the end of the pipeline. Moreover, we

are implicitly estimating the error propagation, as each step is fed the information output from the previous one. These results give thus an idea of the performance of the whole system. Note that the *DLS* 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. This behavior is less evident over the Italian dataset, even tough results confirm the general trend.

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

Table 6: Evaluating the end-to-end chain against the semantic head

In order to provide an even more realistic evaluation of the system, we measured the performance of the system by considering only the prediction over the semantic heads (Table 6). 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 6 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 a F1 score increasing to 93.11%. In addition, the low variance suggests a good stability of the system against 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 dataset is due to two reasons:

		Precision	Recall	F1	RER
			AC		
En	*pLing*	91.04% ± 0.07	91.54% ± 0.07	91.28% ± 0.06	-
	spFeat	92.63% ± 0.05	93.07% ± 0.05	92.83% ± 0.03	17.75%
	Contain	92.72% ± 0.03	93.17% ± 0.05	92.93% ± 0.02	18.83%
	Entity	92.81% ± 0.03	93.26% ± 0.05	93.02% ± 0.02	19.87%
	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	-
	spFeat	83.21% ± 0.44	87.83% ± 0.36	85.39% ± 0.34	2.10%
	Contain	83.42% ± 0.42	88.05% ± 0.35	85.60% ± 0.32	3.54%
	Entity	83.90% ± 0.33	88.58% ± 0.30	86.10% ± 0.25	6.91%
	Ground	84.15% ± 0.33	88.83% ± 0.27	86.35% ± 0.24	8.58%

Table 7: Ablation study of the end-to-end chain against the semantic head

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, being the Italian dataset 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%.

6.5. Ablation study

In order to assess the contribution of the different properties extracted from the Semantic Map, we performed an ablation study of the end-to-end cascade. The performance are measured by considering only the prediction over the semantic head. We tested different configurations of the learning function by incrementally adding the proposed features, finally reaching the complete *Ground* model. The *spFeat* setting refers to a learning function, where spatial features and the Distributional Model DLS are used along with the standard linguistic features; this configuration is then extended with either the Contain-ability property (*Contain*) or the *Entity type* of the grounded entities (*Entity*), as discussed in Section 5. Finally, the *Ground* setting that integrates *all* features has been tested.

Results are shown in Table 7. Over the English dataset we observed that the injection of spatial features reduces the relative error by 17.75% (92.83% F1). This set of features allows to solve most of the PP attachment ambiguities, like the ones mentioned before. Further improvements are obtained with

the *Contain* configuration (18.83% RER - 92.93% F1). This feature has been proven to be useful in the *Closure* frame prediction. In fact, sentences like *"close the jar"* and *"close the door"* generate two different interpretations in terms of frame elements:

$$[close]_{\textbf{\textit{Closure}}}\ [the\ jar]_{\textsc{Containing\_object}}$$

and

$$[close]_{\textbf{\textit{Closure}}}\ [the\ door]_{\textsc{Container\_portal}}$$

Marking the semantic head with the *Contain-ability* property of the grounded object allows to drive the final interpretation towards the correct one. When the *Entity type* of the grounded object is injected as a feature, we get an error reduction of 19.87% (93.02% F1). In this feature space, entities are clustered in categories, explicitly providing further generalization in the learning function.

Conversely, over the Italian dataset we see that spatial properties do not improve consistently the performance, reducing the F1 error of 2.10% (85.39%). This result is probably biased by the language itself, with a small amount of PP attachment ambiguities in the dataset. Instead, a larger contribution is provided by the two domain-dependent features. For example, the *Contain* setting gets an error reduction of 5.47% (85.89% F1), by handling the same ambiguities found in the English dataset. As in the experiment over the English section, the *Entity type* provides a further improvement (6.34% RER - 86.02% F1), due to the generalization of the semantic head. Again, such a discrepancy in the results is mainly due to the different linguistic phenomena therein.

However, in both datasets the best performance are obtained when the full set of features is used, thus providing evidence on (i) the contribution of the different properties, and (ii) the compositionality of the feature spaces.

7. Conclusion

In this work, we presented a comprehensive framework for the definition of robust natural language interfaces for Human-Robot Interaction, specifically designed for the automatic interpretation of spoken commands towards robots in domestic environments. The proposed solution allows to inject domain-dependent and environment-specific 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 [4], we defined a systematic approach to enriching the example representation with additional feature spaces not directly addressable by the linguistic level. Our aim is to leverage the knowledge derived from a semantically-enriched implementation of a robot map (i.e., its Semantic Map), by expressing information about the existence and position of entities surrounding the robot, along with their semantic properties. Observations extracted from the Semantic Map to support 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 in 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*, the Human Robot Interaction Corpus. The corpus, originally composed of examples in English, now contains 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 importantly, it will support the research in grounded natural language interfaces for robots.

The empirical results obtained over both languages are promising, especially when the system is evaluated in a real scenario (end-to-end cascade evaluated against the semantic head); a closer analysis brings about several observations. First, the results confirm the effectiveness of the proposed processing chain, even when only linguistic information is exploited 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 extendable to other languages.

Clearly, there is room to further develop and improve the proposed framework, starting from an extension of HuRIC with additional sentences and semantic features, in order to consider a wider range of robotic actions and

properties. Specifically, future research will focus on the extension of the proposed methodology [4], e.g., by considering spatial relations between entities in the environment or their physical characteristics, such as their color, in the grounding function. In conclusion, we believe that the proposed solution will support further and more challenging research topics in the context of HRI, such as interactive question answering or dialogue with robots.

References

[1] T. Aikawa, C. Quirk, L. Schwartz, Learning prepositional attachment from sentence aligned bilingual corpora, Association for Machine Translation in the Americas, 2003.

[2] K. Church, R. Patil, Coping with syntactic ambiguity or how to put the block in the box on the table, Computational Linguistics 8 (3-4) (1982) 139–149.

[3] T. Wasow, A. Perfors, D. Beaver, The puzzle of ambiguity, Morphology and the web of grammar: Essays in memory of Steven G. Lapointe (2005) 265–282.

[4] E. Bastianelli, D. Croce, A. Vanzo, R. Basili, D. Nardi, A discriminative approach to grounded spoken language understanding in interactive robotics, in: Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI), New York, New York, USA, 2016.

[5] A. Nüchter, J. Hertzberg, Towards semantic maps for mobile robots, Robot. Auton. Syst. 56 (11) (2008) 915–926.

[6] M. Bansal, C. Matuszek, J. Andreas, Y. Artzi, Y. Bisk (Eds.), Proceedings of the First Workshop on Language Grounding for Robotics, Association for Computational Linguistics, Vancouver, Canada, 2017. URL http://www.aclweb.org/anthology/W17-28

[7] G. Salvi, S. Dupont (Eds.), Proceedings GLU 2017 International Workshop on Grounding Language Understanding, Stockholm, Sweden, 2017. doi:10.21437/GLU.2017.

[8] S. Tellex, T. Kollar, S. Dickerson, M. Walter, A. Banerjee, S. Teller, N. Roy, Approaching the symbol grounding problem with probabilistic graphical models, AI Magazine 32 (4) (2011) 64–76.

[9] P. Lindes, A. Mininger, J. R. Kirk, J. E. Laird, Grounding language for interactive task learning, in: Proceedings of the First Workshop on Language Grounding for Robotics, 2017, pp. 1–9.

[10] 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.

[11] J. Thomason, S. Zhang, R. Mooney, P. Stone, Learning to interpret natural language commands through human-robot dialog, in: Proceedings of the 2015 International Joint Conference on Artificial Intelligence (IJCAI), Buenos Aires, Argentina, 2015, pp. 1923–1929.

[12] M. Sahlgren, The word-space model, Ph.D. thesis, Stockholm University (2006).

[13] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, CoRR abs/1301.3781.

[14] E. Bastianelli, D. Croce, R. Basili, D. Nardi, Using semantic models for robust natural language human robot interaction, in: AI* IA 2015, Advances in Artificial Intelligence, Springer International Publishing, 2015, pp. 343–356.

[15] S. Yang, Q. Gao, C. Liu, C. Xiong, S.-C. Zhu, J. 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, Association for Computational Linguistics, 2016, pp. 149–159.

[16] Q. Gao, M. Doering, S. Yang, J. Y. Chai, Physical causality of action verbs in grounded language understanding., in: ACL (1), The Association for Computer Linguistics, 2016.

[17] S. Gella, M. Lapata, F. 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, Association for Computational Linguistics, 2016, pp. 182–192. doi:10.18653/v1/N16-1022.

[18] Y. Berzak, A. Barbu, D. Harari, B. Katz, S. Ullman, Do you see what I mean? visual resolution of linguistic ambiguities, CoRR abs/1603.08079.

[19] M. Alomari, P. Duckworth, M. Hawasly, D. Hogg, A. Cohn, Natural language grounding and grammar induction for robotic manipulation commands (August 2017).

[20] G. Gemignani, R. Capobianco, E. Bastianelli, D. Bloisi, L. Iocchi, D. Nardi, Living with robots: Interactive environmental knowledge acquisition, Robotics and Autonomous Systems 78 (2016) 1–16.

[21] G. Christie, A. Laddha, A. Agrawal, S. Antol, Y. Goyal, K. Kochersberger, D. Batra, Resolving vision and language ambiguities together: Joint segmentation & prepositional attachment resolution in captioned scenes, Computer Vision and Image Understanding 163 (2017) 101 – 112, language in Vision. doi:https://doi.org/10.1016/j.cviu.2017.09.001.

[22] C. Kong, D. Lin, M. Bansal, R. Urtasun, S. 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, IEEE Computer Society, Washington, DC, USA, 2014, pp. 3558–3565. doi:10.1109/CVPR.2014.455.

[23] C. Matuszek, N. FitzGerald, L. S. Zettlemoyer, L. Bo, D. Fox, A joint model of language and perception for grounded attribute learning., in: ICML, icml.cc / Omnipress, 2012.

[24] J. Krishnamurthy, T. Kollar, Jointly learning to parse and perceive: Connecting natural language to the physical world., TACL 1 (2013) 193–206.

[25] Y. Yu, A. Eshghi, O. 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, Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 10–19.

[26] A. Diosi, G. R. Taylor, L. Kleeman, Interactive SLAM using laser and advanced sonar, in: Proceedings of the 2005 IEEE International Con-

ference on Robotics and Automation, ICRA 2005, April 18-22, 2005, Barcelona, Spain, 2005, pp. 1103–1108.

[27] D. L. Chen, R. J. Mooney, Learning to interpret natural language navigation instructions from observations, in: Proceedings of the 25th AAAI Conference on AI, 2011, pp. 859–865.

[28] C. Matuszek, E. Herbst, L. S. Zettlemoyer, D. Fox, Learning to parse natural language commands to a robot control system, in: J. P. Desai, G. Dudek, O. Khatib, V. Kumar (Eds.), ISER, Vol. 88 of Springer Tracts in Advanced Robotics, Springer, 2012, pp. 403–415.

[29] E. Bastianelli, G. Castellucci, D. Croce, R. Basili, D. Nardi, Effective and robust natural language understanding for human-robot interaction, in: Proceedings of ECAI 2014, IOS Press, 2014.

[30] E. Bastianelli, G. Castellucci, D. Croce, R. Basili, D. Nardi, Structured learning for spoken language understanding in human-robot interaction, The International Journal of Robotics Research 36 (5-7) (2017) 660–683.

[31] B. J. Thomas, O. C. Jenkins, Roboframenet: Verb-centric semantics for actions in robot middleware, in: 2012 IEEE International Conference on Robotics and Automation, 2012, pp. 4750–4755. doi:10.1109/ICRA.2012.6225172.

[32] C. J. Fillmore, Frames and the semantics of understanding, Quaderni di Semantica 6 (2) (1985) 222–254.

[33] C. F. Baker, C. J. Fillmore, J. B. Lowe, The berkeley framenet project, in: Proceedings of ACL and COLING, 1998, pp. 86–90.

[34] R. Capobianco, J. Serafin, J. Dichtl, G. Grisetti, L. Iocchi, D. Nardi, A proposal for semantic map representation and evaluation, in: Mobile Robots (ECMR), 2015 European Conference on, IEEE, 2015, pp. 1–6.

[35] D. Pangercic, M. Tenorth, B. Pitzer, M. 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, 2012.

[36] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J.-A. Fernandez-Madrigal, J. González, Multi-hierarchical semantic maps for mobile robotics, in: Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on, IEEE, 2005, pp. 2278–2283.

[37] P. Buschka, A. Saffiotti, A virtual sensor for room detection, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 1, 2002, pp. 637–642.

[38] J. Wu, H. I. Christensen, J. M. Rehg, Visual place categorization: Problem, dataset, and algorithm, in: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, pp. 4763–4770.

[39] O. M. Mozos, H. Mizutani, R. Kurazume, T. Hasegawa, Categorization of indoor places using the kinect sensor, Sensors 12 (5) (2012) 6695–6711.

[40] G. Gemignani, R. Capobianco, E. Bastianelli, D. D. Bloisi, L. Iocchi, D. Nardi, Living with robots: Interactive environmental knowledge acquisition, Robotics and Autonomous Systems 78 (Supplement C) (2016) 1 – 16.

[41] D. Skočaj, M. Kristan, A. Vrečko, M. Mahnič, M. Janíček, G.-J. M. Kruijff, M. Hanheide, N. Hawes, T. Keller, M. 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, IEEE, 2011, pp. 3387–3394.

[42] G. A. Miller, Wordnet: A lexical database for english, Commun. ACM 38 (11) (1995) 39–41.

[43] R. Speer, J. Chin, C. Havasi, Conceptnet 5.5: An open multilingual graph of general knowledge, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA., 2017, pp. 4444–4451.
URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14972

[44] A. Shah, V. Basile, E. Cabrio, S. K. S., Frame instance extraction and clustering for default knowledge building, in: Proceedings of the 1st International Workshop on Application of Semantic Web technologies

in Robotics co-located with 14th Extended Semantic Web Conference (ESWC 2017), Portoroz, Slovenia, May 29th, 2017., 2017, pp. 1–10.

[45] D. S. Chaplot, K. M. Sathyendra, R. K. Pasumarthi, D. Rajagopal, R. Salakhutdinov, Gated-attention architectures for task-oriented language grounding, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[46] Y. Altun, I. Tsochantaridis, T. Hofmann, Hidden Markov support vector machines, in: Proc. of ICML, 2003.

[47] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, Liblinear: A library for large linear classification, Journal of Machine Learning Research 9 (2008) 1871–1874.

[48] S. Filice, G. Castellucci, D. Croce, R. Basili, Kelp: a kernel-based learning platform for natural language processing, in: Proceedings of ACL2015: System Demonstrations, Beijing, China, 2015.

[49] A. Ferraresi, E. Zanchetta, M. Baroni, S. 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, 2008, pp. 47–54.

[50] E. Bastianelli, G. Castellucci, D. Croce, R. Basili, D. Nardi, Huric: a human robot interaction corpus, in: Proceedings of LREC 2014, Reykjavik, Iceland, 2014.

Appendix A. HuRIC: a Corpus of Robotic Commands

The proposed computational paradigms are based on machine learning techniques and strictly depend on the availability of training data. Hence, in order to properly train and test our framework, we developed a collection of datasets that together form the Human-Robot Interaction Corpus (HuRIC)[4], formerly presented in [50].

[4]Available at `http://sag.art.uniroma2.it/huric`. The download page also contains a detailed description of the release format.

HuRIC is based on Frame Semantics 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 for both representing and extracting the interpretation of the command. In fact, it contains information strictly related to Natural Language Semantics and it thus results decoupled from the specific system.

	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 A.8: HuRIC: some statistics

The corpus exploits different situations representing possible commands given to a robot in a house environment. HuRIC is composed of different subsets, characterized by different order of complexity, designed to differently stress a labeling architecture. Each dataset includes a set of audio files representing robot commands, paired with the correct transcription. Each sentence is then annotated with: lemmas, POS tags, dependency trees and Frame Semantics. 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.

HuRIC provides commands in two different languages: English and Italian. While the English subset contains 656 sentences, 241 commands are available in Italian. Almost all Italian sentences are translations of the original commands in English and the corpus keeps also the alignment between those sentences. We believe these alignments will support further researches in further areas, such as in the context of Machine Translation. The number of annotated sentences, number of frames and further statistics are reported in Table A.8. Detailed statistics about the number of sentences for each frame

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 A.9: 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 A.10: Distribution of frames and frame elements in the Italian dataset

and frame elements are reported in Tables A.9 and A.10 for the English and Italian subsets, respectively.

The current release of HuRIC is made available through a novel XML-based format, whose extension is `hrc`. For each command we are able to store: (i) the whole sentence, (ii) the list of the tokens composing it, along with the corresponding lemma and POS tag, (iii) the dependency relations among tokens, (iv) the semantics, expressed in terms of Frames and Frame elements, and (v) the configuration of the environment, in terms of entities populating the Semantic Map (SM). In fact, since in the initial HuRIC version linguistic information were provided without an explicit representation of the environment, we extended the corpus by pairing each utterance with a possible reference environment. Hence, each command is paired with a automatically generated SM, reflecting the disposition of entities matching the interpretation, so that perceptual features can be consistently derived for each command. Extended examples are of the form $\langle s, SM \rangle$. The map generation process has been designed to reflect real application conditions. First, we built a reference Knowledge Base (KB) acting as domain model and containing classes that describe the entities of a generic home environ-

ment. Then, for each sentence s, the corresponding SM is populated with the set of referred entities, plus a control set of 20 randomly-generated additional objects, all taken from the KB. The naming function $\mathcal{LR}$ has been defined simulating the lexical references introduced by a process of Human-Augmented Mapping. The set of possible lexical alternatives (from which such $\mathcal{LR}$ draws) has been designed to simulate free lexicalization of entities in the SM. For every class name in the KB, a range of possible polysemic variations has been defined, by automatically exploiting lexical resources, such as WordNet [42], or by corpus-analysis. The final set has been then validated by human annotators.

CONFLICT OF INTEREST

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us. We confirm that we have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property. In so doing we confirm that we have followed the regulations of our institutions concerning intellectual property.

We understand that the Corresponding Author is the sole contact for the Editorial process (including Editorial Manager and direct communications with the office). He/she is responsible for communicating with the other authors about progress, submissions of revisions and final approval of proofs. We confirm that we have provided a current, correct email address which is accessible by the Corresponding Author and which has been configured to accept email from vanzo@diag.uniroma1.it .

Signed by all authors as follows:

- Andrea Vanzo 06/05/2019
- Danilo Croce 07/05/2019
- Emanuele Bastianelli 06/05/2019
- Roberto Basili 7/5/2019
- Daniele Nardi 06/05/2019