# Language-Based Sensing Descriptors for Robot Object Grounding

Guglielmo Gemignani[1](✉), Manuela Veloso[2], and Daniele Nardi[1]

[1] Department of Computer, Control, and Management Engineering
"Antonio Ruberti", Sapienza University of Rome, Rome, Italy
{gemignani,nardi}@dis.uniroma1.it

[2] Computer Science Department, Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh, PA 15213, USA
veloso@cmu.edu

**Abstract.** In this work, we consider an autonomous robot that is required to understand commands given by a human through natural language. Specifically, we assume that this robot is provided with an internal representation of the environment. However, such a representation is unknown to the user. In this context, we address the problem of allowing a human to understand the robot internal representation through dialog. To this end, we introduce the concept of *sensing descriptors*. Such representations are used by the robot to recognize unknown object properties in the given commands and warn the user about them. Additionally, we show how these properties can be learned over time by leveraging past interactions in order to enhance the grounding capabilities of the robot.

**Keywords:** Sensing descriptors · Human-robot interaction · Natural language processing

## 1 Introduction

One of the main goals of RoboCup@Home is to develop an assistant and companion for humans in domestic settings. The idea is to allow robots to naturally interact with non-expert users in these environments. However, when first interacting with an unknown robot, users may be able to imagine its capabilities, while not knowing how to instruct it. For example, when seeing a manipulator in front of multiple blocks, a user might assume that the robot is able to manipulate them, while being unaware of the commands understood. To this end, several approaches have been proposed to enable untrained users to interact with robots through either constrained or unconstrained natural language.

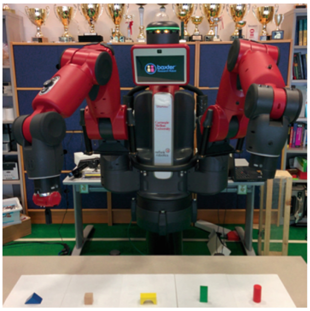
In this paper, we consider the scenario in which a human needs to instruct an autonomous robot through a natural language interface. We assume that this robot is provided with a specific internal representation of the environment

G. Gemignani—contributed to this work while visiting Carnegie Mellon University.

L. Almeida et al. (Eds.): RoboCup 2015, LNAI 9513, pp. 3–15, 2015.
DOI: 10.1007/978-3-319-29339-4_1

that is unknown to the user. For example, a robot might be able to understand colors but not orderings. Also, it may be able to recognize shapes but may not be able to resolve spatial referring expressions. In this scenario, we address the problem of allowing a robot to recognize what object properties can or cannot be grounded with its current sensing capabilities. Moreover, we address the problem of learning new object attributes by exploiting past interactions with the user. While addressing these problems, our goal is to enable an untrained user to understand, through the interaction with the system, which object properties the robot can understand. These interactions can then be used to enhance the grounding capabilities of our robots. Note that in this paper, we will use the term grounding to refer to the concept of "physical symbol grounding" as defined by Vogt [1].

To this end, we contribute a novel approach that enables the robot to recognize unknown objects properties contained in the received commands and warn the user about them. We note that the majority of the techniques proposed in literature make the implicit assumption that if a robot can semantically parse an utterance, then it will be able to ground it. We believe that this assumption may not always hold, since while a robot may be able to correctly parse a sentence and extract its semantics, it may not be able to ground it due to a missing sensing capability. Hence, we internally represent sensing capabilities through *sensing descriptors* and use them to recognize unknown object properties. At this point, the robot can notify the user and request an alternative command. In addition, the robot can learn new object properties by leveraging these interactions with the user. After learning, the robot is able to execute the natural language commands, as in Fig. 1. Our contribution has been used to instruct several robots, including a Baxter manipulator able to perform complex manipulation tasks. In this paper, we describe all the components of our approach along with in depth illustrative examples with the Baxter manipulator robot.



| Commands |
|---|
| - pick up the cubic block |
| - grab the yellow block |
| - touch the second block |
| - point at the left block |
| - take the narrow block |

**Fig. 1.** Baxter manipulator robot used in our experiments and examples of commands that our approach is able to successfully execute.

In the remainder of the paper, we first present an overview of related work, focusing on past research on natural language processing applied to robotic systems. Next, we provide an overview of our natural language approach describing all of our contributions thoroughly. Then, we present an application of the approach to the case of a Baxter manipulator. This setting is then used to quantitatively evaluate the proposed approach. Finally, we conclude with a discussion of our contribution and remarks on future work.

## 2 Related Work

Our research topic is mostly related to the literature on natural language human-robot interaction. Initial studies on natural language understanding can be traced back to SHRDLU [2], a system able to process natural language instructions to perform actions in a virtual environment. Inspired by this system, multiple researchers extended SHRDLU's capabilities into real-world scenarios, soon starting to tackle related problems, including natural language on robotics systems.

Research has applied speech-based approaches to deploy robotic systems in a wide variety of environments. For example, these approaches have been used in manipulators [7–9], aerial vehicles [10], and wheeled platforms [11,12]. Moreover, several prototypes have been developed for social robots carrying out specialized tasks, such as attending as a waiter [13], as a receptionist [15] or as a bartender [14]. Some of these specialized tasks target industrial goals, such as assembly [16], or moving objects [17]. Dialog has also been used to teach robots how to accomplish a given task, such as giving a tour [18], delivering objects [19], or manipulating them [20]. Finally, other related works have combined speech-based approaches with other types of interactions [21,22]. Specifically, in the former work the authors have developed a theory of mind for the interacting user, built upon perspective taking, multi-modal communication, and a symbol grounding capability. Instead, in the latter case, the authors present a multi-modal approach for building on-line a semantic map of the environment.

More recently, several domain-specific systems that allow users to instruct robots through natural language have been presented in literature. For example, Kollar et al. [3] and MacMahon et al. [4] present different methods for following natural language route instructions by decoupling the semantic parsing problem from the grounding problem. In these works, the input sentences are first translated to intermediate representations, which are then grounded into the available knowledge base. Instead, Chernova et al. [5] show how to enable natural language human-robot interaction in a scenario of collaborative human-robot tasks, by data-mining past interactions between humans. Dzifcak et al. [6] address the problem of translating natural language instructions into goal descriptions and actions by exploiting $\lambda-$calculus. However, these approaches are not able to incrementally enhance their natural language understanding from the continuous interaction with the user.

Such a problem has been faced by Kollar et al. [23]. By exploiting the dialog with the user, in this work the authors present a probabilistic approach able to learn referring expressions for robot primitives and physical locations in a map. Our approach is inspired to this latter work. However, we make an additional step forward, assuming the user to be unaware of the capabilities and the internal representation of the robot. With this assumption, we propose an approach for allowing a robot to recognize unknown object properties contained in the received commands and warn the user about them. With this approach, on one hand the user is able to understand over time what a robot can and cannot ground. On the other hand, the robot can leverage past interactions to learn new object properties. The next section describes how our approach can achieve these goals.

## 3 Approach

In this section, first we motivate and introduce the concept of *sensing descriptors*. Next, we present our approach for human-robot natural language interaction based on such a concept. Finally, we show how the system can leverage previous interactions with users to learn previously unknown referring expressions for the objects perceived.

### 3.1 Sensing Descriptors

Usually, when dealing with robots and natural language user commands, a standard processing chain is adopted to decouple the semantic parsing problem from the grounding problem [3,4,19,23]. First, the natural language utterances are converted into text through an automatic speech recognition (ASR) system. Next, the text is converted into a specific representation that captures the semantic meaning of the uttered command. This conversion is carried out either through grammars or probabilistic approaches. The obtained representation is then "contextualized" in the operational environment through a grounding process. The final result is an executable function and a set of parameters passed as input.

In general, during this process each natural language command is grounded through a combination of sensing actions and queries to a given knowledge base. However, this approach does not take into account the sensing capabilities of the robot. In fact, we note that approaches proposed in literature often assume that if the robot can semantically parse an utterance, then it will be able to ground it. However, a robot may be able to correctly parse a sentence and extract its semantics without being able to ground the command due to a missing sensing capability. Hence, we propose to explicitly represent in the knowledge base these capabilities and use them to recognize parts of the commands that could only be grounded through a sensing ability not available to the robot. To this end, we introduce the concept of *sensing descriptor*.

Each sensing operation carried out by a robot can be defined as a function that takes as input a particular type of sensed data and outputs a value expressed in the internal representation of the robot. This value will be an instance of a sensing descriptor. Formally, a sensing operation can be defined as:

$$f_{sensing} : D \rightarrow SD$$

were $D$ is the particular type of data sensed and $SD$ is a specific sensing descriptor. As an example, let's consider the operation of sensing the color of a particular object. The input will be the RGB values of the pixels sensed by a camera. The output will be one or more instances of the sensing descriptor *color* (e.g., $[255, 0, 0]$ or *red* depending on the internal representation of the robot). These sensing descriptors can be used to check if the utterances received from a user can be grounded with the current capabilities of a robot. We perform this check as an intermediate step between the semantic parsing and the grounding process, as explained in the next section.

### 3.2 Human-Robot Natural Language Interaction

Figure 2 shows an overview of our processing chain. Specifically, this processing approach is divided in four consecutive steps. First, speech is converted into text using a free-form speech-to-text engine. Text from speech is confirmed by the user. Thus, without loss of generality, the input of the system is established as natural language text.

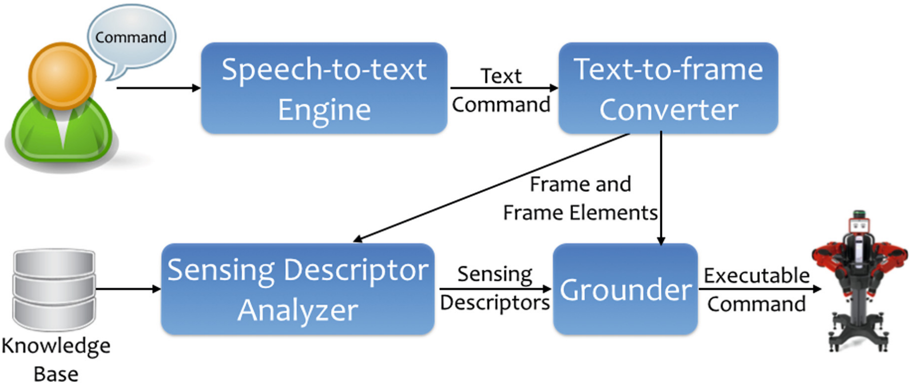


**Fig. 2.** Overview of our natural language processing chain. Instead of directly grounding the frames extracted from the commands, we perform an additional step that analyzes the sensing descriptors included in the frame elements.

Next, the text is converted into a specific representation characterizing the semantics of the sentence. This step is performed through the aid of specific grammars that drive the recognition process by attaching a proper semantic output to each grammar rule. The output has the form of a *semantic frame* representing a "situation" in the world (typically an action) inspired by the notion defined in the *Frame Semantics linguistic* theory [24]. The meaning of each frame can be enriched by semantic arguments, called *frame elements*, that are part of the input sentence. The output of the recognition process is then converted to a parse tree that contains syntactic and semantic information.

This information is used to instantiate a frame, similarly to [25]. As an example, the command "pick up the red block" will be mapped to the GETTING frame. The sub-phrase "the red block" will instead represent the specific frame element THEME, which represents the target of the GETTING action.

At this point, instead of directly grounding the frames in the internal representation of the robot we explicitly represent each sensing descriptor that can be recognized and grounded by the robot, also defining the range of values that it can assume. Formally, in our knowledge base we represent every sensing descriptor $SD_i$ that can be handled by a robot, also representing all its possible known instances $sd_j \in SD_i$. We use these sensing descriptors to check if the obtained frame elements can be grounded with the current sensing capabilities of the robot. Hence, we define *sensing descriptor extractor* a function $\psi$ able to extract from each frame element all the contained instances of sensing descriptors. Formally, if we define $FE$ the frame element type, the *sensing descriptor extractor* can be specified as:

$$\psi : FE \rightarrow \{SD_1, SD_2, ...SD_n\}$$

where $SD_i$ is a specific sensing descriptor extracted from the given frame element.

There are many possible ways to implement this function. In our approach, the sensing descriptor extractor is represented as a parser that exploit grammatical rules to carry out its task. In fact, we note that particular grammar elements are associated to referring expressions that require sensing capabilities to be grounded. Hence, for our specific case, we propose an heuristic rule that selects all the adjectives found in the frame elements. This rule is used to handle element frames such as "the big red and cylindric block" where the word "and" may or may not be used and where the words "big", "red", and "cylindric" need to be extracted. The words extracted represent the sensing descriptor instances that will be checked in the knowledge base. If all the instances are found to belong to a particular sensing descriptor expressed in the knowledge base, the system will proceed to ground the command, otherwise we either leverage dialog or adopt a probabilistic approach to resolve this issue.

### 3.3 Handling Unknown Sensing Descriptors

When an instance of a sensing descriptor is not found in the knowledge base two different scenarios may occur:

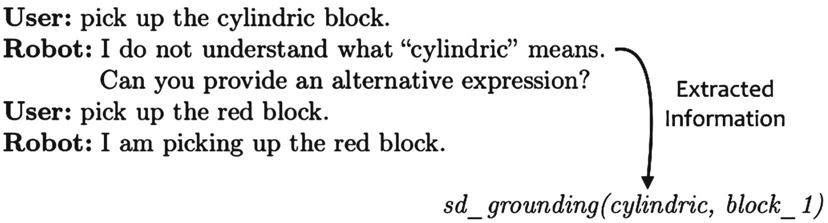
- The referring expression belongs to an unknown sensor descriptor and it has never been used by a user;
- The referring expression belongs to a sensor descriptor not available to the robot but it has been previously used to refer to a particular object.

In the first case, the robot asks the user to provide an alternative referring expression to the object, while keeping track of all the referring expressions used in the different interactions. These expressions are in fact the unknown sensing descriptors found in the frame elements that are not represented in the knowledge base of the robot. Since there is a limited amount of sensing properties that can be expressed, eventually the user will refer to the object in a way that the robot can understand, enabling the robot to associate all the previously used referring expression to the grounding found. To this end, we explicitly represent this association in the knowledge base by using the binary logic predicate *sd_grounding(X, Y)*. In this predicate, $X$ represents the unknown instance of a sensing descriptor, while $Y$ represents the grounding found through the multiple interactions with the user.

For example, let us consider a robot only able to recognize colors. Additionally, let us assume that a user needs to refer to a cylindric red object. At a first interaction a user might refer to the object as “the cylindric block”. When warned by the robot that the term “cylindric” can not be understood, the user will provide a command with an alternative referring expression. Eventually, the user will refer to the object as “the red block”, enabling the robot to correctly ground the expression and assert *sd_grounding(cylindric, block_1)* in his knowledge base. Figure 3 shows an example of a dialog between a manipulator robot and a user that our system is able to understand and the information that the robot is able to extract and store in the knowledge base.

**User:** pick up the cylindric block.
**Robot:** I do not understand “cylindric”.
Are you referring to “blue”?
**User:** No.
**Robot:** Ok, please rephrase the command.
**User:** pick up the red block.

*he_sd_assoc(cylindric, red)*



**User:** pick up the cylindric block.
**Robot:** I do not understand what “cylindric” means.
Can you provide an alternative expression?
**User:** pick up the red block.
**Robot:** I am picking up the red block.

*sd_grounding(cylindric, block_1)*

**Fig. 3.** Example of a dialog between the robot and a user that our system is able to understand and the information extracted and stored in the knowledge base.

To each association instance in the knowledge base, a number is also attached to keep track of how many times the referring expression has been used to refer to a particular object. This counter is needed to handle the alternative scenario that may occur. In this second scenario, a referring expression belonging to an unknown sensor descriptor has been previously used to refer to a particular object. In this case, we adopt a probabilistic approach to ground the expressions. Specifically, if we define KB as the knowledge base available to the robot, R the referring expression being analyzed and G the possible groundings for it, we can obtain the most probable grounding by selecting the one that maximizes Bayes rule:

$$p(G|R;KB) = \frac{p(R|G;KB)\cdot p(G;KB)}{\sum_R p(R|G;KB)\cdot p(G;KB)}.$$

Here, the prior over groundings $p(G;KB)$ is computed by looking at the counts of each element of G in the knowledge base. The other term $p(R|G;KB)$ is instead obtained by counting the number of times a particular referring expression has been used to refer to a particular grounding, and dividing by the overall number of referring expressions used for the same grounding. Formally, if we define *count* the function that returns the number of times that a particular association has been encountered, we can compute $p(R|G;KB)$ as:

$$p(R|G;KB) = \frac{count(association(R,G))}{\sum_R count(association(R,G))}.$$

After having grounded the expressions, we allow the user to give a feedback to the robot to update the counter attached to each association instance. Algorithm 1 reports the overall natural language processing approach. Specifically, the algorithm takes as inputs the natural language command expressed as text and a specific knowledge base. The command is first analyzed to obtain its representation in terms of frames and frame elements (line 3). Next, the sensing descriptor instances are extracted from each frame elements through the sensing descriptor extractor $\psi$ (line 5). Once extracted, the instances are checked against the

**Algorithm 1.** Ground Command

```
Input: Text command $C$, knowledge base $KB$
Data: Frame $f$, set of frame elements $FE$, set of sensing descriptor instances $SD$,
      set of unknown sensing descriptor instances $USD$
Output: Executable action function $\Phi$
1  begin
2      // Extract frames and set of frame elements
3      $f$, $FE \leftarrow$ extractFramesAndFrameElements(C)
4      // Extract the set of sensing descriptor instances
5      $SD \leftarrow \psi(FE)$
6      // Select unknown sensing descriptor instances
7      $USD \leftarrow$ selectUnknownInstances($SD, KB$)
8      if $USD \neq \{\}$ then
9          // Exploit Dialog and Previous Experience to ground command
10         $\Phi \leftarrow$ handleUnknownSensingDescriptors($USD, SD, KB, f, FE$)
11     else
12         // Otherwise normally ground command
13         $\Phi \leftarrow$ ground($f, FE$)
14     return $\Phi$
15 end
```

available knowledge base to find any that cannot be grounded with the current sensing capabilities of the robot (line 7). If an unknown instance is found, the robot exploits dialog and the previous knowledge acquired to assign a grounding to the referring expressions (line 10). Otherwise, the command is grounded into the knowledge base available to the robot to obtain the final executable function (line 13).

## 4 Experimental Evaluation

In this section we describe in detail how the presented approach has been deployed on a Baxter manipulator robot able to manipulate a set of blocks placed in front of it. This setting has been used to quantitatively evaluate our proposed approach. Since the evaluation space of the experiment was large and generating results with humans was extremely time consuming, the experiments were conducted by using a simulator faithful to the chosen setting[1]. A representative sample of the scenarios described in the paper was successfully run on the manipulator interacting with humans, achieving results that are consistent with those reported in the following sections.

### 4.1 Setup

Baxter has two 7 degree of freedom arms, cameras on both arms, and a mounted Microsoft Kinect. Baxter has been programmed to perform the actions touch, grab, move, point to, and push. These primitives are used to manipulate a set of blocks located on a table in front of the robot. The manipulated blocks have different shapes and colors. Additionally, each block has a unique id, associated with a specific QR code. Given this setting, we considered the sensing descriptors shown in Table 1. Specifically, five different blocks were considered:

- A short, wide, triangular, blue block;
- A short, narrow, cubic, brown block;
- A short, wide, bridge-shaped, yellow block;
- A tall, narrow, rectangular, green block;
- A tall, narrow, cylindric, red block.

Additionally, these blocks were associated with the number one through five, respectively. Figure 1 shows the described scenario.

Before accepting commands, the robot was allowed to analyze the scene in order to accumulate knowledge about the operational environment. This knowledge was stored in the form of logic predicates in a knowledge base. The spoken commands given to the robot were converted into text through a free-form ASR[2]. For this particular scenario, a dedicated grammar was developed to convert the natural language commands to the previously described frame representation.

[1] https://github.com/RethinkRobotics/sdk-docs/wiki/Baxter-simulator.

[2] The Google free-form ASR has been used.

**Table 1.** Sensing descriptors considered in the chosen scenario and possible values.

| Sensing descriptors | Possible values |
|---|---|
| color | {blue, brown, yellow, green, red, orange, purple} |
| shape | {triangular, cubic, bridge-shaped, rectangular, cylindric} |
| block id | {first, second, ..., fifth} |
| height | {short, tall} |
| width | {narrow, wide} |
| spatial location | {left, center, right} |

To extract the sensing descriptors from the frame elements, a POS Tagger[3] was used to grammatically analyze the words in the command. Particularly, we adopted the heuristic of extracting the adjectives related to target objects, considering them instances of a specific sensing descriptor. With this approach we were able to allow users to understand how to instruct the robot while interacting with it.

### 4.2 Approach Evaluation

In order to show the effectiveness of our algorithm, we compared our approach with an algorithm commonly used in literature. Specifically, the chosen two-step approach first converts the received commands to frames exploiting grammars. Then, it directly grounds the commands without exploiting any information about sensing descriptors. When the algorithm receives a command that can be grounded to multiple targets (e.g., "touch the narrow block" in this scenario), it selects a random target between the possible ones.

The two approaches have been tested by first generating all the possible commands that can be given to the robot in this setting. Figure 1 shows some example commands generated. Next, 50 commands were randomly chosen and incrementally given in input to the robot. When the robot wasn't able to understand an object attribute, the property was changed with another one not yet used. This process was repeated until the robot understood the command. Such an operation has been carried out for both approaches and averaged for 100 times by varying the number of sensing descriptors known by the robot. For each run we measured the cumulative number of interactions needed to execute all the 50 commands. Figure 4 shows the results obtained in the experiment.

From the graph, it can be noticed that on average our algorithm required significantly less interactions to ground the randomly chosen commands. Moreover, it is worth noticing the effects of the different available levels of information on the two approaches. In fact, when the two robots were capable of understanding and grounding most of the used sensing descriptors, the two approaches had a

[3] We exploited the Stanford POS Tagger to extract the sensing descriptor instances from the frame elements.

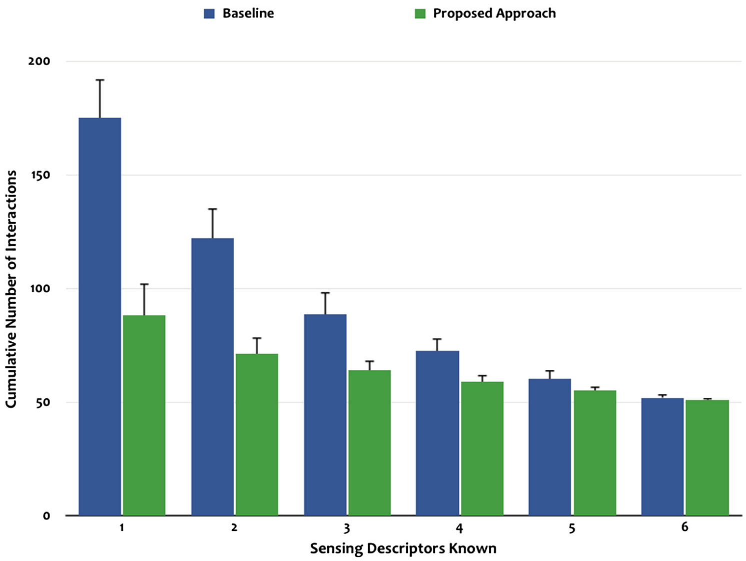


**Fig. 4.** Results for the experiment performed on both processing chains averaged for 100 times by varying the number of sensing descriptors known by the robot.

comparable result. Instead, when a lower amount of information was available, our approach greatly outperformed the other one, leading to a decrease in interactions needed to understand the command, up to approximately 50 % in the chosen scenario.

## 5 Conclusion

In this paper, we considered an autonomous robot provided with an internal representation of the environment, unknown to a user interacting with it through natural language. In this setting, we addressed the problem of allowing humans to understand the internal representation of the robot through dialog. Moreover, we enabled our robot to learn previously unknown object properties leveraging the past interactions with the user. We successfully deployed our approach on a Baxter manipulator robot able to carry out tasks assigned by several users through natural language. Specifically, our experiments report in-detail the performance of our algorithm in this scenario, suggesting an improvement in the grounding effectiveness compared to another commonly used approach.

As a future work, we are studying extensions of the proposed approach. In fact, as a long term goal, we would like to generalize the approach allowing our robots to not only recognize unknown object properties but also every unknown concept contained in the received commands.

## References

1. Vogt, P.: The physical symbol grounding problem. Cogn. Syst. Res. **3**(3), 429–457 (2002)
2. Winograd, T.: Procedures as a representation for data in a computer program for understanding natural language. Technical report (1971)
3. Kollar, T., Tellex, S., Roy, D., Roy N.: Toward understanding natural language directions. In: HRI (2010)
4. MacMahon, M., Stankiewicz, B., Kuipers, B.: Walk the talk: connecting language, knowledge, and action in route instructions. In: AAAI (2006)
5. Chernova, S., Orkin, J., Breazeal, C.: Crowdsourcing HRI through online multiplayer games. In: AAAI Fall Symposium on Dialog with Robots (2010)
6. Dzifcak, J., Scheutz, M., Baral, C., Schermerhorn, P.: What to do and how to do it: translating natural language directives into temporal and dynamic logic representation for goal management and action execution. In: ICRA (2009)
7. Zuo, X., Iwahashi, N., Taguchi, R., Funakoshi, K., Nakano, M., Matsuda, S., Sugiura, K., Oka, N.: Detecting robot-directed speech by situated understanding in object manipulation tasks. In: International Symposium of Robots and Human Interactive Communication (2010)
8. Spangenberg, M., Henrich, D.: Towards an intuitive interface for instructing robots handling tasks based on verbalized physical effects. In: The 23rd IEEE International Symposium on Robot and Human Interactive Communication (2014)
9. Connell, J.H.: Extensible grounding of speech for robot instruction. In: Robots that Talk and Listen (2014)
10. Kollar, T., Tellex, S., Roy, N.: A discriminative model for understanding natural language route directions. In: AAAI Fall Symposium on Dialog with Robots (2010)
11. Kruijff, G., Zender, H., Jensfelt, P., Christensen, H.: Situated dialogue and spatial organization: what, where.. and why. Int. J. Adv. Robot. Syst. **4**(2), 125–138 (2007)
12. Bastianelli, E., Bloisi, D.D., Capobianco, R., Cossu, F., Gemignani, G., Iocchi, L., Nardi, D.: On-line semantic mapping. In: ICAR (2013)
13. Bannat, A., Blume, J., Geiger, J.T., Rehrl, T., Wallhoff, F., Mayer, C., Radig, B., Sosnowski, S., Kühnlenz, K.: A multimodal human-robot-dialog applying emotional feedbacks. In: Ge, S.S., Li, H., Cabibihan, J.-J., Tan, Y.K. (eds.) ICSR 2010. LNCS, vol. 6414, pp. 1–10. Springer, Heidelberg (2010)
14. Stiefelhagen, R., Ekenel, H., Fugen, C., Gieselmann, P., Holzapfel, H., Kraft, F., Nickel, K., Voit, M., Waibel, A.: Enabling multimodal human-robot interaction for the karlsruhe humanoid robot. IEEE Trans. Rob. **23**(4), 840–851 (2007)
15. Nisimura, R., Uchida, T., Lee, A., Saruwatari, H., Shikano, K., Matsumoto, Y.: Aska: receptionist robot with speech dialogue system. In: International Conference on Intelligent Robots and Systems (2002)
16. Foster, M.E., Giuliani, M., Isard, A., Matheson, C., Oberlander, J., Knoll, A.: Evaluating description and reference strategies in a cooperative human-robot dialogue system. In: International Joint Conference on Artificial Intelligence (2009)
17. Tellex, S., Kollar, T., Dickerson, S., Walter, M.R., Banerjee, A.G., Teller, S.J., Roy, N.: Understanding natural language commands for robotic navigation and mobile manipulation. In: AAAI (2011)
18. Rybski, P., Yoon, K., Stolarz, J., Veloso, M.: Interactive robot task training through dialog and demonstration. In: HRI (2007)
19. Gemignani, G., Bastianelli, E., Nardi, D.: Teaching robots parametrized executable plans through spoken interaction. In: AAMAS (2015)

20. Gemignani, G., Klee, S.D., Nardi, D., Veloso, M.: On task recognition and generalization in long-term robot teaching. In: AAMAS (2015)
21. Lemaignan, S., Alami, R.: Talking to my robot: from knowledge grounding to dialogue processing. In: Human-Robot Interaction (2013)
22. Bastianelli, E., Bloisi, D.D., Capobianco, R., Cossu, F., Gemignani, G., Iocchi, L., and Nardi, D.: On-line semantic mapping. In: International Conference on Advanced Robotics (2013)
23. Kollar, T., Perera, V., Nardi, D., Veloso, M.: Learning environmental knowledge from task-based human-robot dialog. In: ICRA (2013)
24. Fillmore, C.J.: Frames and the semantics of understanding. Quad. di Semantica **6**(2), 222–254 (1985)
25. Thomas, B.J., Jenkins, O.C.: Roboframenet: verb-centric semantics for actions in robot middleware. In: ICRA (2012)