LU4R: Adaptive Spoken Language Understanding for Robots

Andrea Vanzo*
Sapienza Università di Roma

Danilo Croce**
Università di Roma, Tor Vergata

Roberto Basili†
Università di Roma, Tor Vergata

Daniele Nardi‡
Sapienza Università di Roma

Service robots are expected to operate in specific environments, where the presence of humans plays a key role. It is thus essential to enable for a natural and effective communication among humans and robots. One of the main features of such robotics platforms is the ability to react to spoken commands. This requires a comprehensive understanding of the user utterance to trigger the robot reaction. Moreover, the correct interpretation of linguistic interactions depends on physical, cognitive and language-dependent aspects related to the environment. In this work, we present the latest version of LU4R - adaptive spoken Language Understanding 4 Robots, a Spoken Language Understanding framework for the semantic interpretation of robotic commands, that is sensitive to the operational environment. The overall system is designed according to a Client/Server architecture in order to be easily deployed in a vast plethora of robotic platforms. Moreover, an improved version of HuRIC - Human-Robot Interaction Corpus is presented. The main novelty presented in this paper is the extension to commands expressed in Italian. In order to prove the effectiveness of such system, we also present some empirical results in both English and Italian computed over the new HuRIC resource.

1. Introduction

One of the most challenging issues that Service Robotics is facing in the recent years is the need of high level interactions and collaborations between humans and robots. In such a robotic context, human language is one of the most natural ways of communication as for its expressiveness and flexibility. However, an effective communication in natural language between humans and robots is challenging even for the different cognitive abilities involved during the interaction. In fact, for a robot to react to a simple command like "*take the pillow on the couch*", a number of implicit assumptions should be met. First, at least two entities, a `pillow` and a `couch`, must exist in the environment and the speaker must be aware of such entities. Accordingly, the robot must have access to an inner representation of the objects, e.g. an explicit map of the

* Dept. of Computer, Control and Management Engineering "Antonio Ruberti" - Via Ariosto 25, 00185 Rome, Italy. E-mail: vanzo@diag.uniroma1.it
** Dept. of Enterprise Engineering - Via del Politecnico 1, 00133 Rome, Italy. E-mail: croce@info.uniroma2.it
† Dept. of Enterprise Engineering - Via del Politecnico 1, 00133 Rome, Italy. E-mail: basili@info.uniroma2.it
‡ Dept. of Computer, Control and Management Engineering "Antonio Ruberti" - Via Ariosto 25, 00185 Rome, Italy. E-mail: nardi@diag.uniroma1.it

environment. Second, mappings from lexical references to real world entities must be developed or made available. In this respect, the *Grounding* process (Harnad 1990) links symbols (e.g. words) to the corresponding perceptual information. Hence, robot interactions need to be *grounded*, as meaning depends on the state of the physical world and the interpretation crucially interplays with perception, as pointed out by psycholinguistic theories (Tanenhaus et al. 1995). The integration of perceptual information derived from the robot's sensors with an ontologically motivated description of the world has been adopted as an augmented representation of the environment, in the so-called *semantic maps* (Nüchter and Hertzberg 2008). In these maps, the existence of real world objects can be associated to *lexical* information, in the form of entity names given by a knowledge engineer or spoken by a user for a pointed object, as in Human-Augmented Mapping (Diosi, Taylor, and Kleeman 2005; Gemignani et al. 2016). While Spoken Language Understanding (SLU) for Interactive Robotics have been mostly carried out over the only evidences specific to the linguistic level (see, for example, (Chen and Mooney 2011; Matuszek et al. 2012)), we argue that such process should be context-aware, in the sense that both the user and the robot live in and make references to a shared environment. For example, in the above command, "*taking*" is the intended action whenever a pillow is actually on the couch, so that "*the pillow on the couch*" refers to a single argument. On the contrary, the command may refer to a "*bringing*" action, when no pillow is on the couch and *the pillow* and *on the couch* correspond to different semantic roles.

We are interested in an approach for the interpretation of robotic spoken commands that is consistent with (*i*) the world (with all the entities composing it), (*ii*) the Robotic Platform (with all its inner representations and capabilities), and (*iii*) the linguistic information derived from the user's utterance.

We foster here the approach presented in (Bastianelli et al. 2016a), where a machine leaning method for Spoken Language Understanding forces the interpretations to be consistent with the environment: this is obtained by extending the linguistic evidences that can be extracted from the uttered commands with perceptual evidences directly derived by the semantic map of a robot. In particular, the interpretation process is modeled as a sequence labeling problem where the final labeler is trained by applying Structured Learning methods over realistic commands expressed in domestic environments, as in (Bastianelli et al. 2017). The resulting interpretations adhere to Frame Semantics (Fillmore 1985): this well-established theory provides a strong linguistic foundations to the overall process while enforcing its applicability, as it is made independent from the vast plethora of existing robotic platforms.

Such methodologies have been implemented in a free and ready-to-use framework, here presented, whose name is *LU4R* - an adaptive spoken **L**anguage **U**nderstanding framework for(**4**) **R**obots. LU4R is entirely coded in Java and, thanks to its Client/Server architectural design, it is completely decoupled from the robot, enabling for an easy and fast deployment on every platform[1].

As the aforementioned approaches relies on realistic data, in this work we also present an extended version of *HuRIC* - a **Hu**man **R**obot **I**nteraction **C**orpus, originally introduced in (Bastianelli et al. 2014). HuRIC is a collection of realistic spoken commands that users might express towards generic service robots. In this resource, each sentence is labeled with morpho-syntactic and syntactic information (e.g. dependency relations, POS tags, . . .), along with its correct interpretation in terms of semantic frames

1 LU4R can be downloaded at `http://sag.art.uniroma2.it/lu4r.html`

(Baker, Fillmore, and Lowe 1998). We present here a new version of HuRIC that has been enhanced in terms of (i) the number of annotated sentences in English and *(ii)* a brand new section, where Italian commands have been added (and aligned) to the already existing English counterparts. At the best of our knowledge this is the first dataset of spoken robotic commands in Italian[2].

The extended version of HuRIC supports a larger and more significant evaluation of LU4R, that highlights its robustness towards commands expressed through the investigated languages. Specifically, we observed very good performances w.r.t. both languages, whose outcomes are encouraging for the deployment of LU4R (and the underlying methods and psycho linguistic assumptions) in realistic applications.

The rest of the paper is structured as follows. Section 2 provides a short survey of existing approaches to SLU in Human-Robot Interaction. Section 3 describes the semantic analysis process that represents the core of the LU4R system. In Section 4, an architectural description of the entire system is provided, as well as an overall introduction about its integration with a generic robot. Section 5 describes the new release of HuRIC, while in Section 6 we demonstrate the applicability of the proposed system in the interpretation of commands in English and Italian, by reporting our experimental results. Finally, Section 7 derives the conclusions.

2. Related Work

In Robotics, Spoken Language Understanding (SLU) has been usually treated by following two orthogonal approaches: grammar-based and data-driven.

Grammar-based systems for speech recognition model language phenomena through the definition of grammars. Moreover, they provide mechanisms to enrich the syntactic structure with semantic information, to build a semantic representation during the transcription process (Bos 2002; Bos and Oka 2007). In (Bastianelli et al. 2016b), SLU supporting manifold robotics tasks is performed jointly with speech recognition, through the definition of ad-hoc grammars. This is possible thanks to the *Speech Recognition Grammar Specification*[3], that allows to inject semantic attachment directly within the grammar specification. Other approaches are based on formal languages, as in (Kruijff et al. 2007), where Combinatory Categorial Grammar (CCG) are applied for spoken dialogues in the context of Human-Augmented Mapping, or exploit template-based algorithms (see (Perera and Veloso 2015)) to extract a semantic interpretation of robotic commands from the corresponding syntactic trees.

Data-driven methods have been also applied to SLU for robotic application. Examples are (MacMahon, Stankiewicz, and Kuipers 2006) and (Chen and Mooney 2011), where the parsing of route instructions is addressed as a Statistical Machine Translation task between the human language and a synthesized robot language. The same approach is applied in (Matuszek, Fox, and Koscher 2010) to learn translation model between natural language and formal descriptions of paths. A probabilistic CCG is used in (Matuszek et al. 2012) to map natural navigational instructions into robot executable commands. The same problem is faced in (Kollar et al. 2010; Duvallet, Kollar, and Stentz 2013), where Spatial Description Clauses are parsed from sentences through sequence labeling approaches. In (Tellex et al. 2011), the authors address natural language instructions about motion and grasping, that are mapped into Generalized Grounding

2 The extended version of HuRIC will be released at `http://sag.art.uniroma2.it/huric.html`
3 `http://www.w3.org/TR/speech-grammar/`

Graphs (G^3). In (Fasola and Matarić 2013a, 2013b), SLU for pick-and-place instructions is performed through a Bayesian classifier trained over a specific corpus. In (Misra et al. 2016), the authors define a probabilistic approach to ground natural language instructions within a changing environment.

2.1 Contribution

On the one hand, LU4R embodies most of the capabilities in terms of linguistic generalization characterizing the presented data-driven approaches. On the other hand, it introduces several novelties that are missing in the existing literature. First, the interpretation is performed and provided in terms of semantic frames, according to the Frame Semantics theory (Fillmore 1985). Hence, the resulting logic form representing the meaning of a command will be supported by a robust linguistic theory. Moreover, as both the proposed semantic parsing approach and the nature of such a theory are domain-independent, the development of a SLU in other domains will depend mostly on the existence of training data. Second, the interpretation process is context-dependent, whenever additional knowledge derived from perception is discriminating against multiple possible interpretations.

3. The Language Understanding Cascade

A command interpretation system for a robotic platform must produce interpretations of user utterances. In this paper, the understanding process is based on the theory of the Frame Semantics (Fillmore 1985); in this way, we aim at giving a linguistic and cognitive basis to the interpretations. In particular, we consider the formalization promoted in the FrameNet (Baker, Fillmore, and Lowe 1998) project, where actions expressed in user utterances can be 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. This information is 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 represents the object that is taken in a *bringing* action.

As an example, let us consider the following sentence: "*bring the pillow on the couch*" ("*porta il cuscino sul divano*", in Italian). This sentence can be intended as a command whose effect is to instruct a robot that, in order to achieve the task, has to: *(i)* move towards a pillow, *(ii)* pick it up, *(iii)* move to the couch and, finally, *(iv)* release the object on the couch. The language understanding cascade should produce its FrameNet-annotated version, that is:

$$[bring]_{Bringing}\ [the\ pillow]_{\text{THEME}}\ [on\ the\ couch]_{\text{GOAL}} \tag{1}$$

or

$$[porta]_{Bringing}\ [il\ cuscino]_{\text{THEME}}\ [sul\ divano]_{\text{GOAL}} \tag{2}$$

whenever the command is expressed through the Italian language.

Semantic frames can thus provide a cognitively sound bridge between the actions expressed in the language and the implementation of such actions in the robot world, namely plans and behaviors.

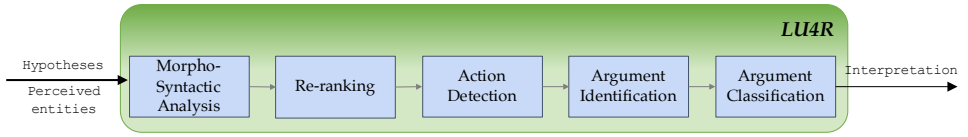

Figure 1
The SLU cascade

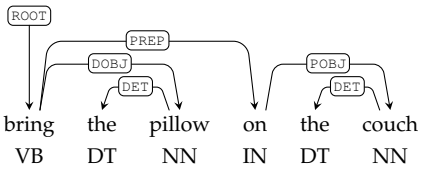

Figure 2
Example of a dependency graph and POS tags associated to "*bring the pillow on the couch*"

The whole SLU process has been designed as a cascade of reusable components, as shown in Figure 1. As we deal with vocal commands, their (possibly multiple) hypothesized transcriptions derived from an Automatic Speech Recognition (ASR) engine constitute the input of this process. It is composed by four modules, whose final output is the interpretation of a utterance, to be used to implement the corresponding robotic actions. First, **Morpho-syntactic and syntactic analysis** is performed over the available utterance transcriptions by applying morphological analysis, Part-of-Speech tagging and syntactic analysis. In particular, dependency trees are extracted from the sentence as well as POS tags, as shown in Figure 2. Then, if more than one transcription hypothesis is available, the **Re-ranking** module can be activated to compute a new ranking of the hypotheses, in order to get the best transcription out of the initial ranking. This module is realized through a learn-to-rank approach, where a Support Vector Machine exploiting a combination of linguistic kernels is applied, according to (Basili et al. 2013). Third, the best transcription is the input of the **Action Detection** (AD) component. The evoked frames in a sentence are detected, along with the corresponding evoking words, the so-called lexical units. Let us consider the recurring sentence: the AD should produce the following interpretation $[bring]_{Bringing}$ *the pillow on the couch*. The final step is the **Argument Labeling**, where a set of frame elements is retrieved for each frame. This process is realized in two sub-steps. First, the *Argument Identification* (AI) finds the spans of all the possible frame elements, producing the following form $[bring]_{Bringing}$ [*the pillow*] [*on the couch*]. Then, the *Argument Classification* (AC) assigns the suitable label (i.e. the frame element) to each span thus returning the final tagging shown in the Example 1.

The AD, AI and AC steps are modeled as a sequential labeling task, as in (Bastianelli et al. 2016a). The Markovian formulation of a structured SVM proposed in (Altun, Tsochantaridis, and Hofmann 2003) is applied to implement the sequential labeler, known as SVM^{hmm}. 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 $\mathbf{x} = (x_1 \dots x_l) \in \mathcal{X}$ of feature vectors $x_1 \dots x_l$, SVM^{hmm} learns a model isomorphic to a k-order Hidden Markov Model, to associate $\mathbf{x}$ with a set of labels $\mathbf{y} = (y_1 \dots y_l) \in \mathcal{Y}$.

A sentence s is here intended as a sequence of words w_i, each modeled through a feature vector x_i and associated to a dedicated label y_i, specifically designed for each interpretation process: in any case, features combine linguistic evidences from a targeted sentences, but also features derived from the semantic map (when available) in order to synthesize information about existence and position of entities around the robot, as discussed in more details in (Bastianelli et al. 2016a) During training, the SVM algorithm associates words to step-specific labels: linear kernel functions are applied to different types of features, ranging from linguistic to perception-based features, and linear combinations of kernels are used to integrate independent properties. At classification time, given a sentence $s = (w_1 \dots w_{|s|})$, the SVM^{hmm} efficiently predicts the tag sequence $\mathbf{y} = (y_1 \dots y_{|s|})$ using a Viterbi-like decoding algorithm.

Notice that both the re-ranking and the semantic parsing phases can be realized in two different settings, depending on the type of features adopted in the labeling process. It is this possible to rely upon linguistic information to solve the given task, or also on perceptual knowledge coming from a semantic map. In the first case, that we call *basic* setting, the information used to solve the task comes from linguistic inputs, as the sentence itself or external linguistic resources. These models correspond to the methods discussed in (Bastianelli et al. 2017; Basili et al. 2013). In the second case, the *simple* setting, when perceptual information is made available to the chain, a context-aware interpretation is triggered, as in (Bastianelli et al. 2016a). Such perceptual knowledge is mainly exploited through a *linguistic grounding* mechanism. This lexically-driven grounding is estimated through distances between filler (i.e. argument heads) and entity names. Such a semantic distance integrates metrics over word vectors descriptions and phonetic similarity. Word semantic vectors are here acquired through corpus analysis, as in Distributional Lexical Semantic paradigms (Turney and Pantel 2010). They allow to map referential elements, such as lexical fillers, e.g. *couch*, to entities, e.g. a `sofa`, by thus modeling synonymy or co-hyponymy. Conversely, phonetic similarities are smoothing factors against possible ASR transcription errors, e.g. *pitcher* and *picture*, allowing to actually cope with spoken language. Once links between fillers and entities have been activated, the sequential labeler is made sensitive to additional features, that inject perceptual information both in the learning and the tagging process, e.g. the presence/absence of referred objects in the environment. As a side effect, the above mechanism provides the robot with the set of linguistically-motivated groundings, that can be potentially used for any further grounding process.

This information can be crucial in the correct interpretation of ambiguous commands, which depends on the specific environmental setting in which the robot operates. A straightforward example is the command *"bring the pillow on the couch in the living room"*. Such a sentence may have two different interpretations, according to the configuration of the environment. In fact, whenever the couch is located into the living room, the goal of the *Bringing* action is the couch and interpretation will be:

$$[\textit{bring}]_{\textit{Bringing}}\ [\textit{the pillow}]_{\textsc{Theme}}\ [\textit{on the couch in the living room}]_{\textsc{Goal}}$$

Conversely, if the couch is outside the living room, it means that probably the pillow is already on the couch. Hence, the interpretation of the sentence will be different, due to different argument spans, and the couch becomes the goal of the *Bringing* action:

$$[\textit{bring}]_{\textit{Bringing}}\ [\textit{the pillow on the couch}]_{\textsc{Theme}}\ [\textit{in the living room}]_{\textsc{Goal}}$$

Such ambiguities are mostly cross-lingual. In fact, this phenomenon can be observed even in the corresponding Italian command *"porta il cuscino sul divano in sala da pranzo"*. However, the proposed approach is robust towards different languages, as the disambiguation of the interpretation depends just on the configuration of the environment and not on the targeted language.

Additional details about the pure linguistic approach can be found in (Bastianelli et al. 2017), whereas (Bastianelli et al. 2016a) provides a detailed description of the context aware SLU process.

4. LU4R - adaptive spoken Language Understanding 4 Robots

The architecture of the LU4R system considers two main actors, as shown in Figure 3: the *Robotic Platform* and the *LU4R chain* (or LU4R), where the processing cascade of the latter component has been introduced in the previous Section.

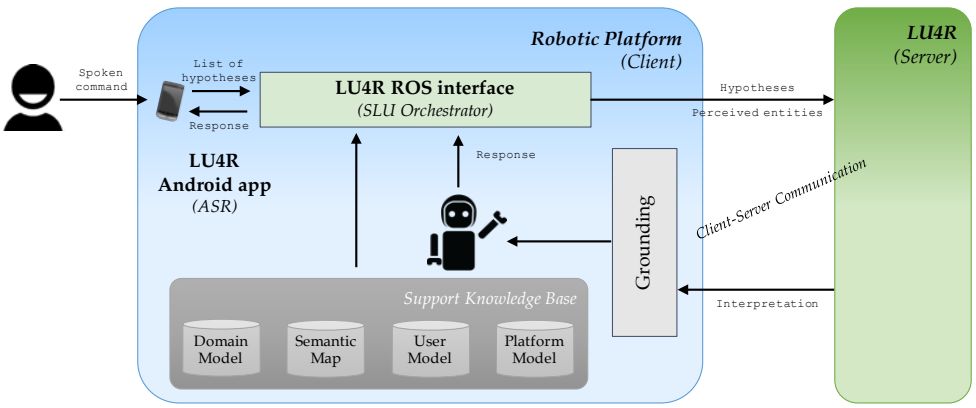

Figure 3
The LU4R architecture

The Client-Server communication schema between LU4R and the Robot allows for the independence from the Robotic Platform, in order to maximize the re-usability and integration in heterogeneous robotic settings. The SLU process exhibits semantic capabilities (e.g. disambiguation, predicate detection or grounding into robotic actions and environments) that are designed to be general enough to be representative of a large set of application scenarios.

It is obvious that an interpretation process must be achieved even when no information about the domain/environment is available, i.e. a scenario involving a *blind* but speaking robot, or when the actions a robot can perform are not made explicit. This is the case when the command "*take the pillow on the couch*" is not paired with any additional information and the ambiguity with respect to the evoked frame, i.e. *Taking* vs. *Bringing*, cannot be resolved. At the same time, LU4R makes available methods to specialize its semantic interpretation process to individual situations where more information is available about goals, the environment and the robot capabilities. These methods are expected to support the optimization of the core SLU process against a specific interactive robotics setting, in a cost-effective manner. In fact, whenever more information about the environment perceived by the robot (e.g. a semantic map) or about its capabilities is provided, the interpretation of a command can be improved by exploiting a more focused scope. That is: whenever the sentence "*take the pillow on the*

couch" is provided along with information about the presence and possible positions of a `pillow` on a `couch`.

In order to better describe the different operating modalities of LU4R, some assumptions toward the Robotic Platform must be made explicit: this will allow to precisely establish functionalities and resources that the robot needs to provide to unlock the more complex processes. These information will be used to express the experience that the robot is able to share with the user (i.e. the perceptual knowledge about the environment where the linguistic communication occurs and some lexical information and properties about objects in the environment) and some level of awareness about its own capabilities (e.g. the primitive actions that the robot is able to perform, given its hardware components). In the following, each component of the architecture in Figure 3 will be discussed and analyzed.

4.1 The Robotic Platform

The LU4R system contemplates a generic Robotic Platform, whose task, domain and physical setting are not necessarily specified. In order to make the SLU process independent from the above specific aspects, we assume that the platform requires, at least, the following modules:

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

In developing LU4R, we implemented both the ASR system and a simple SLU Orchestrator. The ASR is realized by the *LU4R Android app*, exploiting the Android environment, whereas the SLU orchestrator is implemented as a ROS node, through the *LU4R ROS interface*.

Additionally, the optional component *Support Knowledge Base* is expected to maintain and provide the contextual information discussed above. While the discussion about the Robotic Platform is out of the scope of this work, all the other components are hereafter shortly summarized.

LU4R Android app. An ASR engine allows to transcribe a spoken utterance into one or more transcriptions. In the latest release, the ASR is performed through an *ad-hoc* Android application, the LU4R Android app (Fig. 4).

It relies on the official *Google ASR API*[4] and offers valuable performances for an off-the-shelf solution. The main requirement of this solution is that the device hosting the software must feature an Internet connection, in order to provide transcriptions for the spoken utterance. The App can be deployed on both Android smartphones and tablets. In the latter case, even though the communication protocol remains the same, the tablet will be part of the robotic platform. The tablet can be provided with a directional condenser microphone and speakers.

4 `http://goo.gl/4ZkdU`

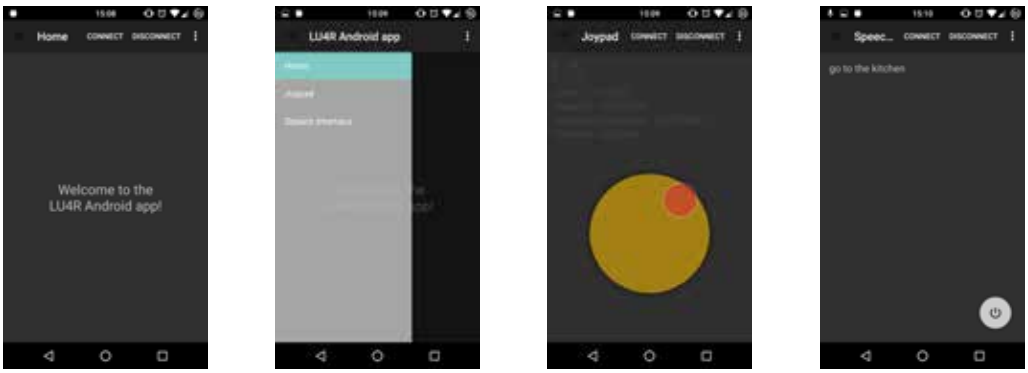

Figure 4
The LU4R Android app

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

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

LU4R ROS interface. The LU4R ROS interface implements a TCP Server for the LU4R Android app, here coded as a ROS node waiting for Client requests. Once a new request is received (a list of transcriptions for a given spoken sentence), this module is in charge of extracting the perceived entities from a structured representation of the environment (here, a sub-component of the Support Knowledge Base) and sending the list of hypothesized transcriptions to LU4R along with the list of the perceived entities.

The communication protocol requires the serialization of such information in two different JSON objects. However, in order to obtain the desired interpretation, only the list of transcription is mandatory. In fact, even though environmental information is essential for the perception-driven chain, whenever it is not provided, the chain operates in a blind setting.

Moreover, this module has been decoupled from the LU4R chain as it can be employed for other purposes, such as tele-operating the robot by means of a virtual joypad coded into the Android App (Fig. 4).

This component, mediating between the LU4R Android App, the LU4R Chain and the Robotic Platform, is provided along with the LU4R system, so that robustness in the communication is guaranteed. Hence, the robotic developers are in charge of: (i) the deployment of the ROS node into the target Robotic System; (ii) the definition of the policies for the acquisition of perceptual knowledge; and (iii) the manipulation of the structure representing the interpretation returned by the LU4R Chain. Even though this module is actually a TCP Server for the LU4R Android App, it represents also the Client interface toward the LU4R Chain.

Grounding and Command Execution. Even though the grounding process is placed at the end of the loop, it is discussed here as it is a component of the Robotic Platform. In fact, this process has been completely decoupled from the SLU process, as it may involve perception capabilities and information unavailable to LU4R or, in general, out

of the linguistic dimension. Nevertheless, this situation can be partially compensated by defining mechanisms to exchange some of the grounding information with the linguistic reasoning component. The grounding carried out by the robot is triggered by a logical form expressing one or more actions through logic predicates, that potentially correspond to specific frames. The output of LU4R embodies the produced logic form: this latter exposes the recognized actions that are then linked to specific robotic operations (primitive actions or plans). Correspondingly, the predicate arguments (e.g. objects and location involved in the targeted action) are detected and linked to the objects/entities of the current environment. A fully grounded command is obtained through the complete instantiation of the robot action (or plan) and its final execution.

4.2 The LU4R Chain

The LU4R component implements the language understanding cascade described in Section 3. It realizes the SLU service as a black-box component, so that the complexity of each inner sub-task is hidden to the user. It is entirely coded in Java and released as a single Jar file.

Morpho-syntactic and syntactic analysis is realized through the Stanford CoreNLP suite (Manning et al. 2014) when English is the targeted language, and the *Chaos* parser (Basili and Zanzotto 2002) for Italian commands. Conversely, the SVM^{hmm} algorithm for the three steps of the semantic analysis (namely, Action Detection, Argument Identification and Argument Classification) is implemented through the KeLP framework (Filice et al. 2015).

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

The initialization phase corresponds to create an instance of the chain, among the ones defined in the previous Section, e.g. either `basic` or `simple`. The `basic` setting does not contemplate perceptual knowledge during the interpretation process. Conversely, the `simple` configuration relies on perceptual information, enabling a context-sensitive interpretation of the command at the predicate level. During the initialization, a specific output format can be chosen, among the available ones. For example, `xdg` is the default output format, where the interpretation is given in the XDG format *eXtended Dependency Graph* and XML compliant container (see (Basili and Zanzotto 2002)). In the `amr` format, the interpretation is given in the *Abstract Meaning Representation* (see (Banarescu et al. 2013)). Finally, `cfr` (*Command Frame Representation*) is a format for the predicates (frames) produced by the chain defined in (Schneider et al. 2014), in the context of RoCKIn competition. The language parameter allows to choose the operating language of LU4R. At the moment, only `en` (English) and `it` (Italian) versions are supported.

Once the service has been initialized, it is possible to start asking for interpreting user utterances. The server thus waits for messages carrying the utterance transcriptions to be parsed. Each sentence here corresponds to a speech recognition hypothesis. Hence, it can be paired with the corresponding transcription confidence score, useful in the re-ranking phase. The body of the message must then contain the list of hypotheses encoded as a JSON array, called `hypotheses`, where each entry is a `transcription` paired with a `confidence`.

Additionally, when the `simple` configuration is selected, the input *can* include the list of `entities` populating the environment the robot is operating into (e.g. name of rooms or furnitures and objects of the rooms), again encoded as a JSON array. Despite of the representation of the environment adopted by the robot, this environment-dependent interpretation process requires the following information for each `entity` "perceived" by the robot:

- the `type` of each entity; it reflects the class to which each specific entity belongs (e.g. it is an object, such as a `table`, `book`, `pillow`, or a location, such as `living_room` or `kitchen`);
- the `preferredLexicalReference` used to refer to a class of objects; it is crucial in order to enable a linguistic grounding between the commands uttered by the user and the entities within the environment. These labels are expected to be provided by the engineer initializing the robot. For example, an entity of the class `couch` can be referred by the string *sofa*. If no label is given, it is derived by the name of the corresponding class, so that *couch* can be used to refer to the objects of the class `couch`;
- in the case the engineer provides more than one label, these can be specified through `alternativeLexicalReference`, as a list of alternative namings for a given entity;
- the position of the each entity is essential to determine shallow spatial relations between entities (e.g. two object are `near` or `far` from each other). To this end, each entity is associated with its corresponding `coordinate` in the world, in terms of planar coordinates (`x`,`y`), elevation (`z`) and `angle` as the orientation.

5. HuRIC 2.0: a multilingual corpus of robotic command

The computational paradigms adopted in LU4R are based on machine learning techniques and depend strictly on the availability of training data. In order to properly train and test our framework, we are developing a collection of datasets that together form the Human-Robot Interaction Corpus[5] (HuRIC), formerly presented in (Bastianelli et al. 2014).

HuRIC is based on Frame Semantics and captures cognitive information about situations and events expressed in sentences. Differently from other corpora for Spoken Language Understanding in Human-Robot Interaction, it is not system or robot dependent both with respect to the kind of sentences and with respect to the adopted formalism. HuRIC contains information strictly related to Natural Language Semantics and it is decoupled from specific systems. The corpus exploits different situations representing possible commands given to a robot in a house environment. HuRIC is composed by different subsets, characterized by different order of complexity and they are designed to stress in different ways a possible architecture. Each dataset includes a set of audio files representing robot commands, paired with the correct transcription.

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

Table 1
HuRIC: some statistics

	English	Italian
Number of examples	656	214
Number of frames	18	14
Number of predicates	767	241
Number of roles	34	27
Predicates per sentence	1.17	1.13
Sentences per frame	36.44	15.29
Roles per sentence	2.04	1.83

Table 2
Distribution of frames and frame elements in the English dataset

Frame	Ex	Frame	Ex	Frame	Ex
Motion	143	***Bringing***	153	***Cotheme***	39
THEME	23	THEME	153	COTHEME	39
GOAL	129	GOAL	95	SPEED	1
DIRECTION	9	AGENT	39	MANNER	9
PATH	9	BENEFICIARY	56	THEME	4
MANNER	4	SOURCE	18	PATH	1
DISTANCE	1	MANNER	1	GOAL	8
AREA	2	AREA	1	AREA	1
SOURCE	1				
Locating	90	***Inspecting***	29	***Taking***	80
PHENOMENON	89	GROUND	28	AGENT	8
GROUND	34	DESIRED_STATE	9	THEME	80
COGNIZER	10	INSPECTOR	5	SOURCE	16
PURPOSE	5	UNWANTED_ENTITY	2	PURPOSE	2
MANNER	2				
Change_direction	11	***Arriving***	12	***Giving***	10
THEME	1	GOAL	11	RECIPIENT	10
DIRECTION	11	PATH	5	THEME	10
ANGLE	3	MANNER	1	DONOR	4
SPEED	1	THEME	1	REASON	1
Placing	52	***Closure***	19	***Change_operational_state***	49
THEME	52	CONTAINER_PORTAL	8	AGENT	17
GOAL	51	AGENT	7	DEVICE	49
AGENT	7	CONTAINING_OBJECT	11	OPERATIONAL_STATE	43
AREA	1	DEGREE	2		
Being_located	38	***Attaching***	11	***Releasing***	9
THEME	38	ITEM	6	THEME	9
LOCATION	34	GOAL	11	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		

Each sentence is then annotated with: lemmas, POS tags, dependency trees[6] 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 4.

With respect to the previous releases, in order to consider further robotic actions, the release of LU4R required an extension of HuRIC in terms of new frames, such as

6 At the moment of writing the dependency trees associated to the Italian Sentences are still under validation

Table 3
Distribution of frames and frame elements in the Italian dataset

Frame	Ex	Frame	Ex	Frame	Ex
Motion	32	***Locating***	27	***Inspecting***	4
GOAL	28	MANNER	2	GROUND	2
MANNER	1	PHENOMENON	27	UNWANTED_ENTITY	2
THEME	3	GROUND	6	INSTRUMENT	1
PATH	2	PURPOSE	1	DESIRED_STATE_OF_AFFAIRS	2
SOURCE	1	DIRECTION	1		
Bringing	59	***Cotheme***	13	***Placing***	18
THEME	60	COTHEME	13	THEME	18
GOAL	26	GOAL	5	GOAL	17
BENEFICIARY	31	MANNER	6	AREA	1
SOURCE	8				
Closure	10	***Giving***	7	***Change_direction***	9
CONTAINING_OBJECT	5	RECIPIENT	6	DIRECTION	9
CONTAINER_PORTAL	6	THEME	7	ANGLE	3
DEGREE	1	DONOR	1	SPEED	1
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				

CHANGE_DIRECTION and, in general, frame elements: at the moment the English subset of HuRiC contains 656 sentences. Most importantly, we extended HuRIC with a first set of 214 commands 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 1. Detailed statistics about the number of sentences for each frame and frame elements are reported in the Tables 2 and 3 for the English and Italian subsets, correspondingly.

The current release of HuRIC is provided with a novel XML-based format, whose extension is `hrc`. For each command we can 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, and *(iv)* the semantics, expressed in terms of Frames and Frame elements.

6. Experimental Evaluation

In order to provide evidences about the effectiveness of the proposed solution, we report here an evaluation of the interpretation process of robotic commands in two languages, i.e. English and Italian, w.r.t the `basic` setting.

Table 4 and 5 show the results obtained over the new version of the Human-Robot Interaction Corpus (HuRIC), presented in Section 5. In fact, the experiments have been performed on both languages, as HuRIC provides commands in both English and Italian. The results, expressed in terms of *Precision*, *Recall* and *F1 measure*, focus on the semantic interpretation process, in particular Action Detection (AD), Argument Identification (AI) and Argument Classification (AC) steps. In fact, F1 scores measure the quality of a specific module. While in the AD step the F1 refers to the ability to extract the correct frame(s) (i.e. robot action(s) expressed by the user) evoked by a

Table 4
English dataset

	Precision	*Recall*	*F1-Measure*
AD	$95.14\% \pm 1.73$	$95.02\% \pm 0.37$	$95.07\% \pm 0.93$
AI	$89.95\% \pm 2.28$	$89.63\% \pm 2.00$	$89.78\% \pm 2.05$
AC	$92.15\% \pm 1.51$	$92.15\% \pm 1.51$	$92.15\% \pm 1.51$

sentence, in the AI step it evaluates to the correctness of the predicted argument spans. Finally, in the AC step the F1 measures the accuracy of the classification of individual arguments.

The experiments have been performed in a random split setting, over 5 iterations. During each iteration, the dataset is shuffled and split into three subsets, containing 70%, 10% and 20% of the data, used as training, tuning and testing set, respectively. In this respect, Table 4 and 5 show also the *standard deviations* among the different iterations.

We tested each sub-module in isolation, feeding each step with gold information provided by the previous step in the chain. Moreover, the evaluation has been carried out considering the correct transcriptions, i.e. not contemplating the error introduced by the Automatic Speech Recognition system. The results over both datasets refer to the `basic` setting of LU4R, that is the configuration in which just linguistic information are exploited.

Results against the commands in English (Table 4) are encouraging for the application of LU4R in realistic scenarios. In fact, the F1 is higher than 95% in the recognition of semantic predicates used to express intended actions (AD). The system is able to recognize the involved entities (AC) with high accuracy as well, with a F1 higher than 92%. 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 total. The most challenging task seems to be the ability to recognize the spans composing a single frame element (AI), where the F1 settles just under the 90% (89.78%).

One of the most frequent errors concerns the ambiguity of the *"take"* verb. In fact, as explained in the previous sections, the interpretation of such verb may be different (i.e. either *Bringing* or *Taking*), depending on the configuration of the environment. As the `basic` 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 that is motivated by the environment. In terms of F1 measure, this issue affects mainly the process of recognizing the argument spans (AI), rather than the ability to identify the action(s) (AD), as for each (possibly) wrong frame, there could be more than two (possibly) wrong arguments. For example, the sentence *"take the pillow on the couch"* will be probably recognized to be a *Taking* action, even though it is labeled as *Bringing*, i.e. *the pillow* and *the couch* are supposed to be far in the environment. While the AD step will receive just one penalty for the wrong recognized action, the AI step is penalized twice, as two arguments were expected by the gold standard annotation, i.e. *the pillow* as THEME and *the couch* as GOAL, instead of one, i.e. *the pillow on the couch* as a single THEME argument. Preliminary experiments in the perception-driven setting seem to show that, whenever such knowledge is in-

jected into the learning process, the system is able to mitigate the error rate over those phenomena.

In addition, small values of standard deviation suggest that the system seems to be rather stable across the different iterations of the experiment and that the results do not depend on specific splits of the entire dataset.

Table 5
Italian dataset

	Precision	*Recall*	*F1-Measure*
AD	$93.59\% \pm 2.81$	$88.63\% \pm 4.25$	$91.01\% \pm 3.21$
AI	$82.80\% \pm 1.38$	$82.50\% \pm 3.47$	$82.64\% \pm 2.34$
AC	$89.93\% \pm 3.83$	$89.93\% \pm 3.83$	$89.93\% \pm 3.83$

The experiments over the Italian dataset reflect the observation that have been pointed out for the English setting. In fact, the system is able to recognize actions (AD) with an F1 measure of 91.01%. Again, valuable performances here suggest that the process of recognizing the intended action(s) is reliable enough to be applicable in real scenarios. As in the English setting, the most challenging step is the Argument Identification, where the F1 measure does not overstep the 83% (82.64%). The results are promising, when compared to the actual size of the dataset. In fact, the classifiers are trained on just the 80% of the entire Italian dataset, i.e. 170 sentences on average. Though lower, the accuracy in recognizing involved entities (AC) is in line with the English experiments, with a F1 score of 89.93%. It seems plausible that the gap in performances and standard deviations with respect to results against the English dataset is mainly due to the reduced size of the dataset.

When looking at the errors, we observe again that the introduction of the perceptual information can be beneficial for the overall task, specially for the AI step. In fact, the command *"porta il bicchiere sul tavolo in cucina"* (i.e. *bring the glass on the table in the kitchen* in English) can not be correctly predicted without information about the involved entities, as two different interpretations are plausible. The intended action correspond to a *Bringing* one in both cases; nevertheless, the involved roles are substantially different. In fact, whenever the referred table (*tavolo*) is inside the kitchen (*cucina*), the table itself represents the goal of the action, whereas if the glass (*bicchiere*) is on the table, this latter is probably outside the kitchen that is, instead, the goal of the action. Hence, the lack of perceptual evidences can play a key role in producing these mis-classifications. Though the F1 measures are not directly comparable as capturing different phenomenon, this behavior could explain the bigger gap that is observed between AD and AI results in the Italian experiment (8.37%) than in the English one (5.29%). Notice that this phenomenon does not affect the AC task. In fact, as we said, during the experimental evaluation each step is fed with gold standard annotations.

At the moment of writing we are pairing each sentence from both sub sets of HuRIC with semantic maps, in order to design proper systematic evaluations also for the `simple`, i.e. context-aware, setting.

7. Conclusions

In this paper, we presented a comprehensive framework for the robust implementation of natural language interfaces for Human-Robot Interaction (HRI). It is specifically

designed for the automatic interpretation of spoken commands towards robots in domestic environments. The solution proposed here relies on Frame Semantics and supports a structured learning approach to language processing able to map individual sentence transcriptions to meaningful commands. An hybrid discriminative and generative learning method is proposed to map the interpretation process into a cascade of sentence annotation tasks.

The overall framework and individual algorithms have been implemented in *LU4R*, a free and ready-to-use Java processing chain, designed for the cost-effective and rapid deployment of language interfaces in a wide range of robotic platforms. By implementing the approach presented in (Bastianelli et al. 2016a),LU4R's command interpretation is made dependent on the robot's environment; in fact the adopted training annotations not only express linguistic evidences from source utterances, but also account for specific perceptual knowledge derived from a reference map. In this way the semantic map aspects useful to interpretation are expressed via feature modeling with the structured learning mechanism applied. Such perceptual knowledge is thus derived from a semantically-enriched implementation of a robot map (i.e. its semantic map): it expresses information about the existence and position of entities surrounding the robot: as this is also available to the user, this information is crucial to disambiguate predicates and role assignments.

The machine learning processes inside LU4R have been trained by using an extended version of *HuRIC*, the Human Robot Interaction Corpus. This corpus, originally composed by example in English, now contains a subset of example in Italian: from the one hand, this novel corpus supports the development of LU4R in the Italian language but, most of all, it will support the research in natural language interfaces for Robots in such language. The empirical results obtained by LU4R over both languages are quite impressive (about 90% of F1 in almost all the evaluations). This *(i)* confirms the effectiveness of the proposed processing chain, *(ii)* the application of the same approach in different languages.

Further effort is required to extend HuRIC with additional sentences, in order to consider a wider range of robotic actions. We are currently working to make it including semantic maps associated to each individual sentence: it will support a systematic evaluation of the interpretation process enhanced with perceptual information. Future research will also focus on the extension of the methodology proposed in (Bastianelli et al. 2016a), e.g. by considering spatial relations between entities in the environment or their physical characteristics, such as their color. In the medium/long term research, we believe that LU4R will support further and more challenging research topics in the context of HRI, such as in interactive question answering or dialogue with robots.

References

Altun, Yasemin, Ioannis Tsochantaridis, and Thomas Hofmann. 2003. Hidden Markov support vector machines. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, Washington D.C., USA, August 21-24.

Baker, Collin F., Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL '98)*, pages 86–90, Montreal, Quebec, Canada, August 10-14.

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

Basili, Roberto, Emanuele Bastianelli, Giuseppe Castellucci, Daniele Nardi, and Vittorio Perera. 2013. Kernel-based discriminative re-ranking for spoken command understanding in hri. In *AI* IA 2013: Advances in Artificial Intelligence*. Springer International, pages 169–180.

Basili, Roberto and Fabio Massimo Zanzotto. 2002. Parsing engineering and empirical robustness. *Natural Language Engineering*, 8(3):97–120, June.

Bastianelli, Emanuele, Giuseppe Castellucci, Danilo Croce, Roberto Basili, and Daniele Nardi. 2014. Huric: a human robot interaction corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014)*, Reykjavik, Iceland, May 26-31.

Bastianelli, Emanuele, Giuseppe Castellucci, Danilo Croce, Roberto Basili, and Daniele Nardi. 2017. Structured learning for spoken language understanding in human-robot interaction. *The International Journal of Robotics Research*, 36(5-7):660–683.

Bastianelli, Emanuele, Danilo Croce, Andrea Vanzo, Roberto Basili, and Daniele Nardi. 2016a. 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, 9-15 July.

Bastianelli, Emanuele, Daniele Nardi, Luigia Carlucci Aiello, Fabrizio Giacomelli, and Nicolamaria Manes. 2016b. Speaky for robots: The development of vocal interfaces for robotic applications. *Applied Intelligence*, 44(1):43–66, January.

Bos, Johan. 2002. Compilation of unification grammars with compositional semantics to speech recognition packages. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING '02)*, volume 1, pages 1–7, Taipei, Taiwan, 26-30 August. Association for Computational Linguistics.

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

Chen, David L. and Raymond J. Mooney. 2011. Learning to interpret natural language navigation instructions from observations. In *Proceedings of the 25th Conference on Artificial Intelligence (AAAI-11)*, pages 859–865, San Francisco, California, USA, August 7-11.

Diosi, Albert, Geoffrey R. Taylor, and Lindsay Kleeman. 2005. Interactive SLAM using laser and advanced sonar. In *Proceedings of the 2005 International Conference on Robotics and Automation*, pages 1103–1108, Barcelona, Spain, April 18-22.

Duvallet, Felix, Thomas Kollar, and Anthony Stentz. 2013. Imitation learning for natural language direction following through unknown environments. In *2013 IEEE International Conference on Robotics and Automation*, pages 1047–1053, Karlsruhe, Germany, May 6-10.

Fasola, Juan and Maja J. Matarić. 2013a. Using semantic fields to model dynamic spatial relations in a robot architecture for natural language instruction of service robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 143–150, Tokyo, Japan, November 3-7.

Fasola, Juan and Maja J. Matarić. 2013b. Using spatial semantic and pragmatic fields to interpret natural language pick-and-place instructions for a mobile service robot. In *Social Robotics: 5th International Conference, ICSR 2013, Bristol, UK, October 27-29, 2013, Proceedings*. Springer International Publishing, pages 501–510.

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

Fillmore, Charles J. 1985. Frames and the semantics of understanding. *Quaderni di Semantica*, 6(2):222–254.

Gemignani, Guglielmo, Roberto Capobianco, Emanuele Bastianelli, Domenico Daniele Bloisi, Luca Iocchi, and Daniele Nardi. 2016. Living with robots. *Robotics and Autonomous Systems*, 78(C):1–16, April.

Harnad, Stevan. 1990. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346.

Kollar, Thomas, Stefanie Tellex, Deb Roy, and Nicholas Roy. 2010. Toward understanding natural language directions. In *Proceedings of the 5th ACM/IEEE International Conference on Human-robot Interaction (HRI '10)*, pages 259–266, Osaka, Japan, March 2-5.

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

MacMahon, Matt, Brian Stankiewicz, and Benjamin Kuipers. 2006. Walk the talk: connecting language, knowledge, and action in route instructions. In *proceedings of the 21st national conference on Artificial intelligence (AAAI'06)*, volume 2, pages 1475–1482. AAAI Press.

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

Matuszek, Cynthia, Dieter Fox, and Karl Koscher. 2010. Following directions using statistical machine translation. In *Proceedings of the 5th ACM/IEEE International Conference on Human-robot Interaction (HRI '10)*, pages 251–258, Osaka, Japan, March 2-5. IEEE Press.

Matuszek, Cynthia, Evan Herbst, Luke S. Zettlemoyer, and Dieter Fox. 2012. Learning to parse natural language commands to a robot control system. In Jaydev P. Desai, Gregory Dudek, Oussama Khatib, and Vijay Kumar, editors, *Experimental Robotics: The 13th International Symposium on Experimental Robotics*, volume 88 of *Springer Tracts in Advanced Robotics*, pages 403–415. Springer.

Misra, Dipendra K., Jaeyong Sung, Kevin Lee, and Ashutosh Saxena. 2016. Tell me dave: Context-sensitive grounding of natural language to manipulation instructions. *The International Journal of Robotics Research*, 35(1-3):281–300.

Nüchter, Andreas and Joachim Hertzberg. 2008. Towards semantic maps for mobile robots. *Robotics and Autonomous Systems*, 56(11):915–926.

Perera, Vittorio and Manuela M. Veloso. 2015. Handling complex commands as service robot task requests. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 1177–1183, Buenos Aires, Argentina, 25-31 July.

Schneider, Sven, Frederik Hegger, Aamir Ahmad, Iman Awaad, Francesco Amigoni, Jakob Berghofer, Rainer Bischoff, Andrea Bonarini, Rhama Dwiputra, Giulio Fontana, Luca Iocchi, Gerhard Kraetzschmar, Pedro Lima, Matteo Matteucci, Daniele Nardi, and Viola Schiaffonati. 2014. The rockin@home challenge. In *Proceedings of the 41st International Symposium on Robotics (ISR/Robotik 2014)*, pages 1–7, Munich, Germany, June 2-3.

Tanenhaus, Michael K., Michael J. Spivey-Knowlton, Kathleen M. Eberhard, and Julie C. Sedivy. 1995. Integration of visual and linguistic information during spoken language comprehension. *Science*, 268:1632–1634.

Tellex, Stefanie, Thomas Kollar, Steven Dickerson, Matthew R. Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. 2011. Approaching the symbol grounding problem with probabilistic graphical models. *AI Magazine*, 34(4):64–76.

Turney, Peter D. and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188, January.