Spoken Language Understanding for Service Robotics in Italian

Andrea Vanzo[1] (✉), Danilo Croce[2], Giuseppe Castellucci[3], Roberto Basili[2], and Daniele Nardi[1]

[1] Sapienza University of Rome, Department of Computer, Control and Management Engineering "Antonio Ruberti"
{vanzo,nardi}@dis.uniroma1.it
[2] University of Roma Tor Vergata, Department of Enterprise Engineering
{croce,basili}@info.uniroma2.it
[3] University of Roma Tor Vergata, Department of Electronic Engineering
castellucci@ing.uniroma2.it

Abstract. Robots operate in specific environments and the correct interpretation of linguistic interactions depends on physical, cognitive and language-dependent aspects triggered by the environment. In this work, we describe a Spoken Language Understanding chain for the semantic parsing of robotic commands, designed according to a Client/Server architecture. This work also reports a first evaluation of the proposed architecture in the automatic interpretation of commands expressed in Italian for a robot in a Service Robotics domain. The experimental results show that the proposed solution can be easily extended to other languages for a robust Spoken Language Understanding in Human-Robot Interaction.

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

1 Introduction

End-to-end communication in natural language between humans and robots is challenging for the deep interaction of different cognitive abilities. For a robot to react to a user command like "*porta il libro sul tavolo nel laboratorio*"[1], a number of implicit assumptions should be met. First, at least three entities, `libro` (book), `tavolo` (table) and `laboratorio` (laboratory), must exist in the environment and the speaker must be aware of such entities. Hence, the robot must have access to an inner representation of the objects, e.g., an explicit map of the environment. Second, mappings from lexical references to real world entities must be made available. *Grounding* [1], here, should correspond to the explicit linking of symbols (e.g., words) to the information perceived about the context. Spoken Language Understanding (SLU) for interactive dialogue systems acquires

[1] In English, "*bring the book on the table in the laboratory*".

a specific nature when applied to Interactive Robotics. Linguistic interactions are context-aware in the sense that both the user and the robot access and make references to the environment (i.e., entities of the real world). In the above example, whenever a table is actually in the laboratory, the GOAL of the action referred by the verb "*portare*" ("*to bring*") is [*sul tavolo nel laboratorio*], i.e., the book has to be brought on the table in the laboratory. On the contrary, if there are no tables in the laboratory, [*sul tavolo*] is needed to locate the book nearby the robot and the GOAL refers to [*nel laboratorio*], i.e., the book is on a table and it has to be brought in the laboratory. Hence, robot interactions need to be *grounded*, as meaning depends on the state of the physical world and interpretation crucially interacts with perception, as pointed out by psycho-linguistic theories [2]. The integration of perceptual information derived from the robot's sensors with an ontologically motivated description of the world provides an augmented representation of the environment, called *semantic map* in [3]. 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 [4]. While SLU for Interactive Robotics has been mostly carried out over the evidences specific to the linguistic level, e.g., in [5,6,7], we argue that such process should be accomplished in a harmonized and coherent manner. In fact, SLU has been already addressed in other works (see, for example, [8,9]) where perceptual knowledge is neglected in disambiguating among the structures produced by a linguistic parser.

This paper presents a processing chain for the interpretation of spoken commands. This chain is based on the approach proposed in [10] that integrates both linguistic and perceptual information to realize a context-aware interpretation of robotic commands. In particular, the interpretations coherently express constraints about the world (with all the entities composing it), the Robotic Platform (with all its inner representations and capabilities) and the pure linguistic level. Moreover, we present an experimental evaluation of the proposed chain over a dataset of commands in Italian, to validate its effectiveness with respect to different languages. To the best of our knowledge this is the first work addressing SLU of robotic commands in Italian Language. Preliminary results confirm the effectiveness of the adopted approach even in Italian: a first processing chain in Italian can be in fact obtained by annotating about 10 sentences representing typical ways to express a robotic command in a domestic environment.

In Section 2, the overall processing work-flow is introduced. In Section 3, we provide an architectural description of the chain, as well as an introduction about its integration with a generic robot. In Section 4, we present the experimental results of the proposed system over a dataset of Italian commands. Finally, in Section 5 we derive the conclusions.

2 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 [11]; 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 [12] 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: "*porta il libro sul tavolo*". This sentence can be intended as a command (in Italian), whose effect is to instruct a robot to bring a book on a table. The language understanding cascade should produce its FrameNet-annotated version, that is:

$$[porta]_{Bringing}\ [il\ libro]_{\textsc{Theme}}\ [sul\ tavolo]_{\textsc{Goal}} \tag{1}$$

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.

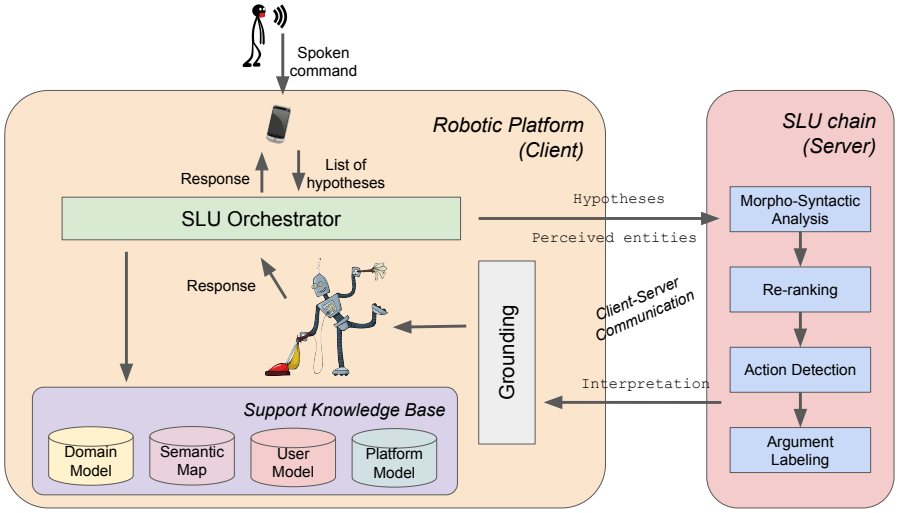

Fig. 1. Overall Architecture of the SLU chain

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, later adopted to implement the corresponding robotic actions. First, **Morpho-syntactic analysis** is performed over the available utterance transcriptions by applying morphological analysis and Part-of-Speech tagging. In our evaluations, an off-the-shelf tool is adopted for this module, i.e., the *Chaos* parser [13]. 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 [14]. 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 one recurring sentence: the AD should produce the following interpretation $[porta]_{Bringing}$ *il libro sul tavolo*. The AD step is realized through a sequential labeling approach: each token of a sentence is labeled through a Markovian Support Vector Machine [15] with respect to the possible frames evoked by the token, according to [7]. 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. Then, the *Argument Classification* (AC) assigns the suitable label (i.e. the frame element) to each span thus producing the final tagging shown in the example 1. The Argument Labeling phase is realized through a sequential labeling algorithm similar to the one of the previous phase. Here, each token of a sentence is associated to one (or none) frame element of the detected frame.

Notice that both the re-ranking and the semantic parsing phases can be realized in two different settings. They can either exploit only linguistic information to solve the given task, or they can embed also perceptual knowledge coming from a semantic map into the process. In the first case, 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 [7,14]. In the second case, robot's *perceptual* information can be made available to the chain, as in [10]. In this way, perceptual information such as the existence of grounded entities, as well as spatial relations among them, can be made available during the interpretation 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.

3 The overall Architecture

The architecture of the proposed system involves two main actors, as shown in Figure 1: the *Robotic Platform* and the *Spoken Language Understanding Chain*

(or SLU Chain), where the main concepts of the latter component have been introduced in the previous section.

A Client-Server communication schema between the SLU chain and the Robot allows maintaining a perspective on the SLU Chain, which strictly emphasizes the independence from the Robotic Platform, in order to maximize the re-usability and integration in heterogeneous robotic settings. The SLU process we propose 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, that we would call an *unaware* linguistic robot. This is the case when the command "*porta il libro sul tavolo nel laboratorio*" is not paired with any additional information and the ambiguity with respect to the argument spans, i.e., [*il libro sul tavolo*]$_{\text{THEME}}$ [*nel laboratorio*]$_{\text{GOAL}}$ vs. [*il libro*]$_{\text{THEME}}$ [*sul tavolo nel laboratorio*]$_{\text{GOAL}}$, cannot be resolved. At the same time, the platform makes available methods to specialize its semantic interpretation process to individual situations where more information is available about goals, environment and capabilities of the robot. These methods are expected to support the optimization of the core SLU platform 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 context. It means that whenever the sentence "*porta il libro sul tavolo nel laboratorio*" is provided along with information about the presence and possible positions of a `table` referred by the word *tavolo* in a `laboratory` (*laboratorio*) the system should be able to detect and disambiguate the intended action. In order to better describe the different operating modalities of the proposed SLU Chain, some assumptions about 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 1 will be discussed and analyzed[2].

3.1 The Robotic Platform

The SLU Chain contemplates a generic Robotic Platform, whose task, domain and physical setting are not necessarily specified. In order to make the SLU

[2] A more detailed description of the proposed SLU Chain along with usage instructions can be found at `http://sag.art.uniroma2.it/sluchain.html`

Chain independent from the above specific aspects, we will assume that the platform requires at least the following modules:

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

Additionally, an optional component, i.e., the *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.

ASR system. An ASR engine allows to transcribe a spoken utterance into one or more possible transcriptions. In the actual release, the ASR is here performed through an *ad-hoc* Android application. In fact, it relies on the official *Google ASR API*[3] 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. This App can be deployed on both Android smartphones and tablets.

Once a new sentence is uttered by the user, this component outputs a list of candidate hypothesis transcriptions. The communication with the entire system is realized through TCP Sockets. In this setting, the Android ASR App implements a TCP Client, feeding the SLU Chain with lists of hypotheses.

SLU Orchestrator. The SLU Orchestrator implements a TCP Server for the Android App, here coded as a ROS[4] node waiting for Client requests. Once a new request arrives (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 of sending the list of hypothesized transcriptions to the SLU Chain along with the list of the perceived entities.

The communication protocol requires the serialization of such information in two different JSON objects. In order to obtain the desired interpretation, only the list of transcription is mandatory. In fact, even though environment information is essential for the perception-driven chain, whenever it is not provided, the chain operates in a blind setting. The SLU orchestrator has been decoupled from the SLU 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. To this end, it can be personalized (or even replaced with a new one), by adding further functionalities and features, provided that the communication protocol is respected. The orchestrator, managing the communication between the Android App, the SLU Chain and the Robotic Platform, is provided along with the SLU Chain, so that robustness in the communication is guaranteed. In this way, the robotic developers are in charge of: (i) the ROS node of the target Robotic System; (ii) the definition of the policies for the acquisition of perceptual knowledge; and

[3] `http://goo.gl/4ZkdU`

[4] `http://www.ros.org/`

(iii) the manipulation of the structure representing the interpretation returned by the SLU Chain. In fact, the SLU orchestrator, besides acting as TCP Server for the Android App, represents also the Client interface toward the SLU Chain.
Grounding and Command Execution. Even though the grounding process is placed at the end of the SLU processing chain, it is discussed here as it represents part of the Robotic Platform. In fact, grounding has been completely decoupled from the SLU Chain, as it may involve perception capabilities and information unavailable to the SLU Chain 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. However, grounding is always carried out on board of the robot, as it represents the most general situation.

The grounding carried out by the robot is triggered by a logical form expressing one or more actions through logic predicates, which potentially correspond to specific frames. The output of the whole SLU process embodies the produced logic form. This latter exposes: the recognized actions that are thus linked to specific robotic operations (primitive actions or plans); the predicate arguments (e.g., objects and location involved in the targeted action) detected and linguistically linked to the objects/entities of the current environment. A fully grounded command is thus obtained where possible through the complete instantiation of the robot action (or plan) and its final execution.

3.2 The SLU Chain

The SLU Chain component implements the language understanding cascade described in Section 2. It realizes the SLU service as a black-box component, so that the complexity of each inner sub-task is hidden to the robotic engineer. The service is realized through a server accepting connections on a predefined port. It is entirely coded in Java and released as a single Jar file, along with the required folders containing linguistic models, configurations files and other resources. Hence, it can be run through command line, so that it is easier to integrate it within any architecture. Operationally, the chain takes three parameters as inputs: *type* of the chain (`basic` or `simple`), *output format* (`XDG`, `AMR` or `TAB`) and *listening port* (e.g., `9090`). The first parameter defines the type of the chain going to be initialized. While `basic` refers to a setting where only linguistic information is employed, i.e., the *blind* situation, `simple` refers to the more complex chain, where perceptual features are taken into account in the interpretation process.

The second parameter specifies the desired output format. The type `XDG` refers to a Java data structure specifically devoted to the overall linguistic analysis of a command, called *eXtendend Dependency Graph*, whose details can be found in [13]. The type `AMR` refers to the *Abstract Meaning Representation*, a semantic representation language proposed in [16]. This formalism allows to express semantics, neglecting both the original sentence and its syntactic structure. Given the sentence "*porta il libro sul tavolo*", the corresponding AMR format is:

```
(t1 / porta-Bringing
    :Theme (l1 / il libro)
    :Goal (t2 / sul tavolo)
)
```

4 Experimental evaluation

In this section, we report a preliminary experimental evaluation of the Spoken Language Understanding (SLU) Chain presented in this paper applied in the interpretation of commands in Italian. The experiments have been designed in order to verify the robustness of the adopted SLU solution in the robotic context with different languages. The evaluation reported here extends the experiments already carried out in [10], where the above SLU chain has been evaluated against commands in English.

Frame	Examples	Frame	Examples
Being_in_category	2	***Inspecting***	4
Category	1	Desired_state_of_affairs	3
Item	2	Purpose	2
		Instrument	2
Being_located	13	***Following***	12
Theme	12	Cotheme	6
Location	11	Goal	3
Cotheme	1	Manner	7
Time	1	Source	1
Bringing	39	***Motion***	34
Goal	12	Goal	28
Beneficiary	2	Source	1
Theme	34	Path	2
Source	6	Manner	1
Place	2	Theme	1
		Direction	2
		Degree	2
Change_direction	6	***Entering***	1
Direction	6	Goal	1
Change_operational_state	15	***Releasing***	2
Device	15	Theme	2
Closure	5	***Searching***	29
Containing_object	1	Manner	2
Place	2	Phenomenon	29
Instrument	4	Ground	8
Manner	1	Degree	1
Placing	21	***Taking***	28
Theme	18	Theme	28
Goal	20	Source	16
Source	1	Place	4

Table 1. Distribution of frames and frame elements in the Italian dataset

We produced an Italian dataset by translating a significant subset of English commands from the HuRIC corpus [17] already used in [10]. Each translated command is also manually labeled according to the Frame Semantics theory, that provides a semantic layer for the command interpretation process, as discussed in Section 2: semantic frames and frame elements are here used to represent the meaning of commands, reflecting the actions a robot can accomplish in a home environment. To this end, we considered the same set of FrameNet-inspired semantic frames adopted in [10], that act as language independent primitives for the robot's possible actions. Linguistic information required for each processing step has been extracted by using the *Chaos* parser [13]. The dataset is composed of 188 different commands, whose actions are represented by 14 different frames. It contains 211 annotated frames (i.e., almost $1,12$ annotated frame per sentence) and 304 annotated roles (i.e., 1.62 role per sentence). The composition of the dataset in terms of number of sentences evoking each frame and number of annotated examples for each role is reported in Table 1.

In the following experiments, we first evaluated each sub-module in the chain separately, then we focused in the overall processing chain, thus considering the error propagated during the analysis.

4.1 Evaluation of the individual modules in the SLU chain

The proposed SLU Chain has been first evaluated by considering in isolation each sub-modules discussed in Section 2, i.e., the Action Detection (AD), Argument Identification (AI) and Argument Classification (AC) sub-modules. To this end, we invoke each module by assuming that the information provided by the previous step in the chain is always correct. Moreover, the evaluation has been carried out considering the correct transcriptions, i.e., not contemplating the error introduced by the Automatic Speech Recognition (ASR) system. In this way, we focus on the errors of the SLU Chain and avoid the bias introduced by the ASR system. Given the limited size of the training material, experiments have been performed in a leave-one-out setting, i.e., each example is in turn removed from the dataset and it is adopted as test example: the remaining examples are adopted to train the chain while performances are derived by averaging results across the entire dataset. In these experiments, we do not consider perceptual information derived from the environment where the command has been pronounced, as these new commands in Italian are not completely aligned with the maps used in [10], yet. Results reported here are thus comparable with those obtained in [10], when a pure linguistic approach is addressed. We report the performance measures, in terms of Micro Precision (P), Recall (R) and F1-Measure (F1), with respect to the single sub-module.

In the AD phase, P, R and F1 measure the system effectiveness in correctly recognizing the frame(s) for each sentence, i.e., the robotic action(s) in our scenario. In the AI phase, they quantify the system ability in recognizing the exact boundaries of each argument in the frame. This means that *every* token (i.e., span) of every argument must be properly detected. In the AC phase, they are a measure of the correctness of the role label assignment to each span.

Action Detection			Argument Identification			Argument Classification		
P	R	F1	P	R	F1	P	R	F1
86,39%	78,57%	82,29%	81,82%	77,23%	79,46%	84,49%	84,49%	84,49%

Table 2. Experimental evaluation over the Italian dataset of each single sub-module in terms of Precision (P), Recall (R) and F1-Measure (F1)

The results for the three phases are reported in Table 2. Even though this setting does not reflect a real operating scenario, where the performance drop is due to the error propagation during the semantic understanding process, this experiment provides an interesting food for thought about the complexity of each task. First, the most challenging task seems to be the AI, whose F1 is the lowest among the three phases, i.e., 79.46% of F1 is obtained. On the contrary, AD and AC obtain higher F1 scores ($82,28\%$ and $84,49\%$, respectively).

These results are in general lower with respect to the results obtained over the entire HuRIC, in [10]. In fact, while in the AD task over the English dataset the system obtains a F1 score of 94.67%, the same evaluation over Italian commands achieve a F1 score of 82.29%. A similar drop of performances is observed both in AI and AC: in the AI task the English dataset allows the system to reach 90.74% in the F1 score, while in AC the score is 94.93%. These results are compared, respectively, with 79.46% and 84.49% obtained over Italian commands. We speculate that such drop is mainly due to the size of the involved dataset. In fact, while Italian data count a total of 188 commands, the evaluation of the system over the English language has been carried out over 527 commands.

However, this empirical investigation confirms the overall trend of performances, with the Argument Identification task the most complex one, and proves that the proposed system can be robustly extended to other languages.

4.2 Evaluation of the whole process

In a second experiment, we analyze the error propagation through the whole SLU Chain. To this end, the performances measured in each step take into account the errors made by the previous one. As an example, let us consider the AI sub-module, where the identification of the frame elements does depend on the frames assigned in the previous step, i.e. the AD sub-module. If an action is not detected, its corresponding argument will be not identified neither. This issue is considered in the evaluation of the next steps, while it has been neglected in the previous evaluations. This setting thus reflects a more realistic operating scenario, where the performance drop is due to the error propagation. Again, we report the performance measures in terms of Micro Precision (P), Recall (R) and F1-Measure (F1), with respect to each single sub-module.

The results for the three phases are reported in Table 3. As expected, a performance drop across the SRL steps has been obtained, when possible incorrect

Action Detection			Argument Identification			Argument Classification		
P	R	F1	P	R	F1	P	R	F1
86,39%	78,57%	82,29%	85,27%	63,04%	72,49%	79,02%	58,61%	67,30%

Table 3. Experimental evaluation over the Italian dataset of the whole processing chain in terms of Precision (P), Recall (R) and F1-Measure (F1)

information has been provided at each step. While the AD phase gets the same results (i.e., the non-gold setting for the AD is not provided as it is the first step in the proposed chain) if we consider the AI phase, the F1 score of 79.46% in Table 2 measured with gold-standard information drops to 72.49% of Table 3, when enabling error propagation by feeding non-gold information through modules. A performance drop is observed in the AC phase when compared with the gold setting, where we measure a F1 score of 67.30% against the 84.49% of the gold setting. However, the overall chain seems to be quite robust to the error propagation as, given the previous measurement made in isolation, a lower result was expected. In fact, the coarse multiplication of the F1 scores obtained in the single steps, i.e., 82.29%, 79, 46% and 84, 49%, corresponds to about 55% of F1. Such experimental results suggest that the proposed solution is promising for the development of SLU chains in different languages, as these results have been obtained only labeling about 13 sentences per frame, i.e., robot command.

5 Conclusion

In this paper, we presented a SLU processing chain focused on the problem of interpreting commands in the Mobile Service Robotics domain. The proposed solution relies on well-known theories, such as Frame Semantics and Distributional Semantics and leverages Machine Learning algorithms to support the interpretation of commands. These characteristics enabled for a more robust interpretation of the sentences against language variability. Moreover, even though the SLU Chain is completely decoupled from the Robotic Platform, the final interpretation has been tied to the environment surrounding the robot: it will allow to inject perceptual knowledge into the feature modeling process, as foreseen by the English chain ([10]). In order to prove the effectiveness of the proposed tool, we conducted some experiments on a real robotic scenario by addressing a new language, i.e., interpreting commands in Italian. Preliminary evaluations show promising results that can be obtained by only labeling a very limited set of examples, i.e., about 10 sentences for each robot action, and by relying only on pure linguistic information. Further evaluations will take into consideration both an extended version of the Italian dataset and the alignment of its commands with perceptual knowledge. We expect that the proposed SLU chain can support the development of natural language interfaces for Human Robot Interaction for further languages than English and Italian.

References

1. Harnad, S.: The symbol grounding problem. Physica D: Nonlinear Phenomena **42**(1-3) (1990) 335–346
2. Tanenhaus, M., Spivey-Knowlton, M., Eberhard, K., Sedivy, J.: Integration of visual and linguistic information during spoken language comprehension. Science **268** (1995) 1632–1634
3. Nüchter, A., Hertzberg, J.: Towards semantic maps for mobile robots. Robot. Auton. Syst. **56**(11) (2008) 915–926
4. Diosi, A., Taylor, G.R., Kleeman, L.: Interactive SLAM using laser and advanced sonar. In: Proc. of the 2005 International Conference on Robotics and Automation. (2005) 1103–1108
5. Chen, D.L., Mooney, R.J.: Learning to interpret natural language navigation instructions from observations. In: Proc. of the 25th AAAI Conference. (2011) 859–865
6. Matuszek, C., Herbst, E., Zettlemoyer, L.S., Fox, D.: Learning to parse natural language commands to a robot control system. In Desai, J.P., Dudek, G., Khatib, O., Kumar, V., eds.: ISER. Volume 88 of Springer Tracts in Advanced Robotics., Springer (2012) 403–415
7. Bastianelli, E., Castellucci, G., Croce, D., Basili, R., Nardi, D.: Effective and robust natural language understanding for human-robot interaction. In: Proceedings of ECAI 2014, IOS Press (2014)
8. Tellex, S., Kollar, T., Dickerson, S., Walter, M., Banerjee, A., Teller, S., Roy, N.: Approaching the symbol grounding problem with probabilistic graphical models. AI Magazine **32**(4) (2011)
9. Matuszek, C., FitzGerald, N., Zettlemoyer, L.S., Bo, L., Fox, D.: A joint model of language and perception for grounded attribute learning. In: ICML, icml.cc / Omnipress (2012)
10. Bastianelli, E., Croce, D., Vanzo, A., Basili, R., Nardi, D.: A discriminative approach to grounded spoken language understanding in interactive robotics. In: Proc. of the 25th IJCAI, New York. (2016)
11. Fillmore, C.J.: Frames and the semantics of understanding. Quaderni di Semantica **6**(2) (1985) 222–254
12. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The berkeley framenet project. In: Proceedings of ACL and COLING. (1998) 86–90
13. Basili, R., Zanzotto, F.M.: Parsing engineering and empirical robustness. Nat. Lang. Eng. **8**(3) (June 2002) 97–120
14. Basili, R., Bastianelli, E., Castellucci, G., Nardi, D., Perera, V.: Kernel-based discriminative re-ranking for spoken command understanding in hri. In: AI* IA 2013: Advances in Artificial Intelligence. Springer International (2013) 169–180
15. Altun, Y., Tsochantaridis, I., Hofmann, T.: Hidden Markov support vector machines. In: Proc. of ICML. (2003)
16. Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., Schneider, N.: Abstract meaning representation for sembanking. In: Proc. of the 7th Linguistic Annotation Workshop and Interoperability with Discourse, Sofia, Bulgaria, ACL (August 2013) 178–186
17. Bastianelli, E., Castellucci, G., Croce, D., Basili, R., Nardi, D.: Huric: a human robot interaction corpus. In: Proc. of LREC 2014, Reykjavik, Iceland (may 2014)