Perceptually Informed Spoken Language Understanding for Service Robotics

Emanuele Bastianelli[1], **Danilo Croce**[2], **Andrea Vanzo**[3], **Roberto Basili**[2], **Daniele Nardi**[3]
[1]DICII, [2]DII, University of Rome Tor Vergata, Rome, Italy
[3]DIAG, Sapienza University of Rome, Rome, Italy
bastianelli@ing.uniroma2.it, {croce,basili}@info.uniroma2.it, {vanzo,nardi}@dis.uniroma1.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 introduce a Spoken Language Understanding chain for semantic parsing of robotic commands. It has been designed according to a Client/Server architecture in order to be easily integrated with the vast plethora of robotic platforms. The robustness of the proposed system against the language variability and its adaptability with respect to the environment surrounding a robot is demonstrated over real scenarios, in the Service Robotics domain.

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 "*take the book on the table*", a number of implicit assumptions should be met. First, at least two entities, a `book` and a `table`, 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 environment. Second, mappings from lexical references to real world entities must be developed or made available. *Grounding* here [Harnad, 1990] links symbols (e.g., words) to the corresponding perceptual information. Spoken Language Understanding (SLU) for interactive dialogue systems acquires a specific nature, when applied in 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, "*taking*" is the intended action whenever a book is actually on the table, so that "*the book on the table*" refers to a single argument. On the contrary, the command may refer to a "*bringing*" action, when no book is on the table and *the book* and *on the table* correspond to different semantic roles (i.e., THEME and GOAL). 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 [Tanenhaus *et al.*, 1995].

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* [Nüchter and Hertzberg, 2008]. 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 spoken by a user for a pointed object, as in Human-Augmented Mapping [Diosi *et al.*, 2005; Bastianelli *et al.*, 2013]. While SLU for Interactive Robotics have been mostly carried out over the only evidences specific to the linguistic level, e.g., in [Chen and Mooney, 2011; Matuszek *et al.*, 2012; Bastianelli *et al.*, 2014], we argue that such process should proceed in a harmonized and coherent manner. All linguistic primitives, including predicates and semantic arguments, correspond to perceptual counterparts, such as plans, robot's actions or entities involved in the underlying events.

In [Bastianelli *et al.*, 2016], a SLU process that integrates perceptual and linguistic information has been proposed. This process allows to produce sentence interpretations that 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. A discriminative approach based on the Markovian formulation of Support Vector Machines is adopted, where grounded information is directly injected within the learning algorithm, showing that the integration of linguistic and perceptual knowledge improves the quality and robustness of the overall interpretation process.

In this paper, we present the *Spoken Language Understanding Chain* based on the model proposed in [Bastianelli *et al.*, 2016]. This chain is fully implemented in JAVA, and is released according to a Client/Server architecture, in order to decouple the chain from the specific robotic architecture that will use it: while the Robotic Platform represents the *Client*, the Spoken Language Understanding Chain is the *Server*. The communication between these modules is realized through a simple and dedicated protocol: the chain receives as input one or more transcriptions of a spoken command and produces an interpretation that is consistent with a linguistically-justified semantic representation, coherent with the perceived environment (i.e., FrameNet). The rest of the paper is structured as

follows. In Section 2, the overall processing workflow is introduced. In Section 3, we provide an architectural description of the chain, as well as an overall introduction about its integration with a generic robot. Section 4 describes the communication protocol of the proposed architecture. In Section 5, we demonstrate the robustness of the implemented solution when used by a real robot. Finally, in Section 6 we derive the conclusion and we discuss future extensions to the SLU Chain.

2 The Language Understanding Cascade

The understanding process proposed here has the goal of producing an interpretation of an user utterance in terms of Frame Semantics [Fillmore, 1985], in order to give a linguistic and cognitive basis to the interpretation. Specifically, we consider the formalization adopted in the FrameNet [Baker *et al.*, 1998] database. According to such theory, actions expressed in user utterances can be modeled as *semantic frames*. These are micro-theories about real world situations, e.g., the action of *taking*. Each frame specifies also the set of participating entities, called *frame elements*, e.g., the THEME representing the object that is taken during the *Taking* action. For example, for the sentence

"take the book on the table"

whose corresponding parsed version can be

$$[\textbf{\textit{take}}]_{Taking}\ [\textit{the book on the table}]_{\text{THEME}} \quad (1)$$

In a robotic perspective, semantic frames 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.

The SLU process has been synthesized in a processing chain based on a set of reusable components. It takes as input one or more hypothesized utterance transcriptions, depending on the employed third party Automatic Speech Recognition (ASR) engine. As one can see in Figure 1, the chain is composed by four modules:

- **Morpho-syntactic analysis** is performed over every available utterance transcription, applying Part-of-Speech tagging and syntactic parsing, providing morphological and syntactic information, essential for further processing.
- If more than one transcription hypothesis is available, a **Re-ranking** module can be activated to evaluate a new sorting of the hypotheses, in order to get the best transcription out of the original ranking.
- The selected transcription is the input of the **Action Detection** (AD) component. Here all the frames evoked in a sentence are detected, according to their trigger lexical units. For example, given the above example, the AD would produce the following interpretation: $[\textbf{\textit{take}}]_{Taking}$ *the book on the table*.
- The final step is the **Argument Labeling** (AL). Here a set of frame elements is retrieved for each frame detected during the AD step. Such process is, in turn, realized in two sub-steps. First, the *Argument Identification* (AI) aims at finding the spans of all the possible frame elements. Then, the *Argument Classification* (AC) assigns the suitable frame element label to each span identified during the AI, producing the final tagging shown in 1.

An off-the-shelf tool is used for the morpho-syntactic analysis, namely the Stanford CoreNLP Library [Manning *et al.*, 2014]. Re-ranking is performed using a learn-to-rank approach, where a Support Vector Machine exploiting a combination of linguistic kernels is applied, according to [Basili *et al.*, 2013]. The AD and AL steps together perform a general *semantic parsing* phase, for which a custom statistical semantic parser built according to [Croce *et al.*, 2012] has been adopted. Every step of the process is modeled as a sequential labeling problem, where a label specific to the current step is associated to every word composing a sentence. The Markovian formulation of a structured SVM proposed in [Altun *et al.*, 2003] is applied to solve the different sequential labeling problems.

Both the re-ranking and the semantic parsing phases can work 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, all the information used to solve the task comes from linguistic inputs, as the sentence itself or external linguistic resources. Notice how these models corresponds to different methods more deeply discussed in [Basili *et al.*, 2013; Bastianelli *et al.*, 2014]. In the second case, perceptual information is made available to the chain by a semantic map that represents the robot's perception of the environment, as in [Bastianelli *et al.*, 2015a; Bastianelli *et al.*, 2016]. In this way, perceptual information such as existence of grounded entities, as well as spatial relations among these, is made available during the interpretation process. This allows to better interpret ambiguous commands, whose correct interpretation depends also on the environmental setting. In order to extract such a perceptual knowledge, the system relies on a *linguistic grounding* mechanism [Bastianelli *et al.*, 2015a; Bastianelli *et al.*, 2015c], that exploits the Distributional Semantics paradigms. It allows to link labels representing entity names in a semantic map with words used in a sentence to refer to such entities, e.g., the word *book* with the entity name `book`. By applying this mechanism, a set of possible candidate groundings over the semantic map is obtained for a given word. As a side effect, the produced interpretation can thus provide the robot with a set of potential groundings obtained only by looking at linguistic aspects (i.e., entity names and words). Such set can be used by the robot as additional information for any further grounding process.

3 The overall Architecture

The architecture of the proposed system considers 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.

The Client-Server communication schema between the SLU chain and the Robot allows maintaining a perspective

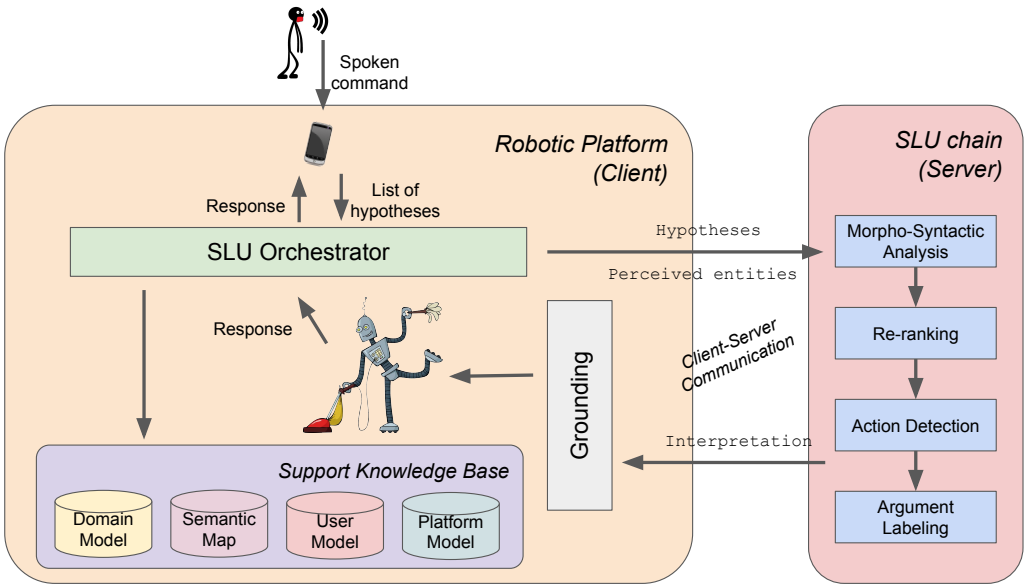

Figure 1: Overall Architecture of the SLU chain

on the SLU Chain that strictly emphasizes the independence from the Robotic Platform, in order to maximize the reusability 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 "*take the book on the table*" 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, the platform 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 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 scope. That is: whenever the sentence "*take the book on the table*" is provided along with information about the presence and possible positions of a `book` on a `table`.

In order to better describe the different operating modalities of the proposed SLU Chain, 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 1 will be discussed and analyzed.

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 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 Engine;
- a Physical Robot.

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.

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*[1] and offers valuable performances for an off-the-shelf solution. The main requirement of this solution is that

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

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 hypothesized 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. To this end, a SLU Orchestrator has been integrated in the loop, acting as the TCP Server.

SLU Orchestrator. The SLU Orchestrator implements a TCP Server for the Android App, here coded as a ROS node[2] 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 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 as provided in more details in the next Section. However, 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.

Moreover, this module has been decoupled from the SLU Chain as it can be employed for other purposes, such as teleoperating the robot by means of a virtual joypad coded into the Android App. To this end, this component can be personalized (or even replaced with a new one), by adding further functionalities and features, provided that the communication protocol is respected.

This component, 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 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 SLU Chain. Even though this module is actually a TCP Server for the Android App, it 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 loop, it is discussed here as it represents part of the Robotic Platform. In fact, this process 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, that potentially correspond to specific frames. The output of the SLU Chain 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.

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

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 user. 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 input parameters: *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 data structure specifically devoted to the overall linguistic analysis of a command, called *eXtendend Dependency Graph* [Basili and Zanzotto, 2002] (an example is not reported due to lack of space). The type `AMR` refers to the *Abstract Meaning Representation*, a semantic representation language proposed in [Banarescu *et al.*, 2013]. This formalism allows to express semantics, neglecting both the original sentence and its syntactic structure in the final representation. Hence, given the sentence "*take the book on the table*", the corresponding AMR format, when the *Taking* frame is evoked, is:

```
(t1   / take-Taking
      :Theme (b1 / the book on the table)
)
```

4 Communication Protocol Description

Functional-wise, the SLU 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 output an interpretation of them. The communication between the client of the service (the Robotic Platform) and the SLU Chain follows a protocol that is described in this section, along with its usage and main functionalities. The usage of the SLU Chain passes through two phases: the *initialization phase*, where the chain is run and initialized, and a *service phase*, where the chain is ready to

receive requests. The rest of this Section provides a clear description of the several modes of operation, together with the messages syntax and format required by the protocol.

4.1 Initialization Phase

Before accepting sentences for their interpretation, the service needs to be run and initialized. Such operation corresponds to initialize an instance of the chain, among the ones defined in the previous Section, e.g., either `basic` or `simple`. Launching the service is quite simple. In fact, it can be performed through the following command line:

`java -jar slu-chain.jar <type><output><port>`

where:

- `<type>` is the parameter that specifies the type of the chain to launched (i.e., `basic` or `simple`);
- `<output>` specifies the output format, among the available ones (i.e., `amr`, `xdg`, or `conll`)
- `<port>` set the listening port of the service.

For example, the command

`java -jar slu-chain.jar simple amr 9090`

runs the chain which exploits perceptual information on the port 9090 and returns the interpretation in the Abstract Meaning Representation format.

4.2 Service Phase

Once the service has been initialized, it is possible to start asking for interpreting user utterances. The server thus waits for messages carrying the utterance transcriptions to be parsed, according to the following protocol.

Spoken Language Understanding The sentences over which we want to have an interpretation must be sent to this service. Each sentence here corresponds to a speech recognition hypothesis. Hence, it can be paired with the corresponding transcription confidence score, useful in the reranking 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` according to the following syntax:

```
{"hypotheses" : [
  {"transcription" : "take the book on the table",
  "confidence" : "0.9",
  "rank" : "1"},
  ...
]}
```

Additionally, when the `simple` chain is selected, it requires the list of entities populating the environment the robot is operating into[3], in order to enable perceptual features. This additional information must be passed as the `entities` parameter in the following JSON format:

```
{"entities" : [
  {"atom" : "book1",
  "type" : "book",
  "preferredLexicalReference" : "book",
  "alternativeLexicalReferences" : [
    "volume","manual",...],
  "coordinate" : {
  "x" : "13.0",
  "y" : "6.5",
  "z" : "3.5",
  "angle" : "3.5"}},
  ...
]}
```

The service can be invoked with a `HTTP POST` request, as follows:

`http://127.0.0.1:9090/service/nlu`
`POST parameters: hypo = {"hypotheses" : [...]}`
`entities = {"entities" : [...]}`

Accepting/Rejecting interpretations After an interpretation has been received by the robot, a confirmation message should be sent back to notify the correct reception and execution of the corresponding command. Hence, the chain expects a plain confirm message at the end of each plan executed by the robot. As for the interpretation service, this acknowledgment is a simple `HTTP POST` request:

`http://127.0.0.1:9090/service/confirm`

When an interpretation can not be executed for several reasons, the robot is enabled to send a rejection message to the server. Such a message is, again, a `HTTP POST` request:

`http://127.0.0.1:9090/service/reject`

The chain thus stores the rejected interpretations that can be used in the future to re-train the statistical models underlying the chain, e.g., by adopting online learning schemas.

5 Experiments on a real robot

In this Section, we show the integration of this tool on a real robot. Moreover, we provide some qualitative evaluations of its application in real scenarios[4].

The robot employed during the experiments is a modified version the Videre Erratic (Fig. 2), equipped with different trays, each of which hosting sensors (e.g., RGB-D, Laser scanners,...) and additional components. The semantic map is acquired in a previous stage with the same robot through Human-Augmented Mapping [Diosi *et al.*, 2005; Bastianelli *et al.*, 2013]. Speech recognition is performed through the Google Speech APIs, available within the Android environment. Such a mobile application communicates with the SLU Orchestrator, by sending the hypothesized transcriptions and receiving the interpretation of the command, displayed on the smartphone screen. The SLU Orchestrator is a `ROS` node coded in `Python`.

We run different tests, aimed at showing the effectiveness of the SLU Chain with respect to different real scenarios. We considered only commands involving *"taking"* and *"bringing"* actions that present diverse ambiguities the proposed system is able to resolve, even though the robot does not feature a manipulator. Hence, in our system, such commands can actually be executed with the help of an user, as proposed in the Symbiotic Autonomy approach.

[3] Notice that if the entity list is empty, the `simple` chain operates as the `basic` one.

[4] A video showing the tests can be found here: `http://sag.art.uniroma2.it/demo-software/slu_chain/`

Figure 2: The robot used for the experimental evaluation

Robustness of the linguistic analysis The first situation taken into account refers to the lexical generalization problem. In fact, the same entity can be evoked by different referring expressions. For instance, while a user may use the term "*book*", another one may call it "*volume*" to refer to the same entity. To this end, we setup an environment with a book and a table. Additionally, we included some other objects to simulate a real scenario. All these information are encoded as a semantic map, where the book is referred by the word "*book*". When the user says the command "*take the volume on the table*" through the Android application, the list of transcriptions is sent to the SLU Orchestrator. This latter extracts the entities from the semantic map and encodes both the JSON parameters to be sent to the chain, i.e., list of transcriptions and entities. The resulting interpretation links the word "*volume*" to the book entity and the robot performs the action of taking the book in the environment.

Perception-driven Action Detection In this situation, we show how the system we propose is robust in terms of action disambiguation. In fact, a verb may evoke different actions, depending on some spatial relations among the involved entities. This is the case of the command "*take the book on the table*", where the verb "*take*" may be interpreted as either *Taking* or *Bringing* action. Hence, the semantic map plays a key role in interpreting such a sentence. To prove the robustness of the interpretation, we tested two situations.

In a first setting, we placed a book on a table and asked for the interpretation of the sentence "*take the book on the table*". Before the interpretation of the command, the SLU Orchestrator extracts the entities from the semantic map that are sent to the SLU Chain along with the hypothesized transcription. The resulting interpretation is the following:

```
(t1   / take-Taking
      :Theme (b1 / the book)
)
```

stating that the book to be taken is the one placed on the table.

In the second setting, the book and the table are set far from each other. In this case, the SLU Chain interprets the command as a *Bringing* action

```
(t1   / take-Bringing
      :Theme (b1 / the book)
      :Goal (t2 / on the table)
)
```

that is "the book has to be brought to the table". Hence, the table represents here the goal of the action.

Perception-driven Argument Labeling In the last tested scenario, we aimed at showing how the SLU Chain is able to recognize the arguments of an action, according to the configuration of the environment. In this experiment, we setup two different environments. The former displays a table in a room (i.e., here, the laboratory) with a book placed on it. The latter presents a book that is still on the table, but they are both outside of the above room. We tested the command "*bring the book on the table in the laboratory*" for both the scenarios. Even though the resulting interpretation in the AMR format seems to be the same

```
(t1   / bring-Bringing
      :Theme (b1 / the book)
      :Goal (t2 / on the table)
)
```

if we look at the labeling of the arguments, we notice a substantial difference among the two cases. In fact, in the first scenario the chain is able to identify the arguments as follow

$[\textbf{\textit{bring}}]_{Bringing}$ [*the book*]$_{\text{THEME}}$ [*on the table in the laboratory*]$_{\text{GOAL}}$

whereas in the second setting the interpretation of the arguments changes substantially, becoming

$[\textbf{\textit{bring}}]_{Bringing}$ [*the book on the table*]$_{\text{THEME}}$ [*in the laboratory*]$_{\text{GOAL}}$

and proving that the final interpretation accounts for the spatial relations of the entities involved in the sentence.

6 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. This has been obtained by relying on Distributional lexical Models to generalize and catch the semantics hidden behind words' surface form. 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, by injecting perceptual knowledge into the feature modeling process. Such a knowledge is extracted by synthesizing the information collected from a semantic map into a feature space. In order to prove the effectiveness of the proposed tool, we conducted some experiments on a real robot by addressing different scenarios, confirming our insights. Each tested setting successfully showed the main contributions that this SLU Chain brings to the community, namely: robustness against language variability, perception-driven Action Detection and perception-driven Argument Labeling. However, at this stage the proposed linguistic chain does not take into account re-ranking approaches for the hypothesized transcriptions, even though we proposed possible solutions in our previous works [Bastianelli *et al.*, 2015b; Vanzo *et al.*, 2016]. Moreover, our system still neglects a set of more complex interactions with the robot as well as techniques to improve the accuracy of the models by refining them as the interaction proceeds. All these aspects will constitute the topic of our future research.

References

[Altun *et al.*, 2003] Yasemin Altun, I. Tsochantaridis, and T. Hofmann. Hidden Markov support vector machines. In *Proc. of ICML*, 2003.

[Baker *et al.*, 1998] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The berkeley framenet project. In *Proceedings of ACL and COLING*, pages 86–90, 1998.

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

[Basili and Zanzotto, 2002] Roberto Basili and Fabio Massimo Zanzotto. Parsing engineering and empirical robustness. *Nat. Lang. Eng.*, 8(3):97–120, June 2002.

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

[Bastianelli *et al.*, 2013] Emanuele Bastianelli, Domenico Daniele Bloisi, Roberto Capobianco, Fabrizio Cossu, Guglielmo Gemignani, Luca Iocchi, and Daniele Nardi. On-line semantic mapping. In *Advanced Robotics (ICAR), 2013 16th International Conference on*, pages 1–6, Nov 2013.

[Bastianelli *et al.*, 2014] Emanuele Bastianelli, Giuseppe Castellucci, Danilo Croce, Roberto Basili, and Daniele Nardi. Effective and robust natural language understanding for human-robot interaction. In *Proceedings of ECAI 2014*. IOS Press, 2014.

[Bastianelli *et al.*, 2015a] Emanuele Bastianelli, Danilo Croce, Roberto Basili, and Daniele Nardi. Using semantic maps for robust natural language interaction with robots. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[Bastianelli *et al.*, 2015b] Emanuele Bastianelli, Danilo Croce, Roberto Basili, and Daniele Nardi. Using semantic maps for robust natural language interaction with robots. In *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association*, pages 1393–1397, 2015.

[Bastianelli *et al.*, 2015c] Emanuele Bastianelli, Danilo Croce, Roberto Basili, and Daniele Nardi. Using semantic models for robust natural language human robot interaction. In *AI* IA 2015, Advances in Artificial Intelligence*, pages 343–356. Springer International Publishing, 2015.

[Bastianelli *et al.*, 2016] Emanuele Bastianelli, Danilo Croce, Andrea Vanzo, Roberto Basili, and Daniele Nardi. A discriminative approach to grounded spoken language understanding in interactive robotics. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York*, 2016.

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

[Croce *et al.*, 2012] D. Croce, G. Castellucci, and E. Bastianelli. Structured learning for semantic role labeling. *Intelligenza Artificiale*, 6(2):163–176, 2012.

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

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

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

[Manning *et al.*, 2014] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

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

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

[Tanenhaus *et al.*, 1995] M. Tanenhaus, M. Spivey-Knowlton, K. Eberhard, and J. Sedivy. Integration of visual and linguistic information during spoken language comprehension. *Science*, 268:1632–1634, 1995.

[Vanzo *et al.*, 2016] Andrea Vanzo, Danilo Croce, Emanuele Bastianelli, Roberto Basili, and Daniele Nardi. Robust spoken language understanding for house service robots. In *Proceedings of the 17th International Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2016, Konya, Turkey*, 2016.