Natural Language Processing for Human-Robot Interaction

Domingo Senise de Gracia
MSc in Artificial Intelligence
Universidad Politécnica de Madrid
Boadilla del Monte, 28660 Madrid (Spain)
domingo.senise@haitta.com

ABSTRACT

It would simply reinforce what is evident, if it were stated: "Robots are becoming an essential part in our daily lives". Currently the use of robots in several and different industries is widespread. Nonetheless, how do we communicate and interact with robots? If robots are playing gradually a more important role in our day-to-day routines, we will have to develop humanlike dialogue-processing mechanisms to ease that interaction.

In this paper the use of natural language processing techniques applied to human-robot interaction is analyzed from three different perspectives: Firstly, it is exposed the design of flexible dialogue-based robotic systems, from data collected in human interaction experiments, in the context of a search task[1]. Secondly, techniques that allow humans to teach new high-level actions to robots through step-by-step natural language instructions are explained[3]. And thirdly, a method to create mappings from sentences -using shallow semantic parsing- to expected robot actions is examined[4]. The three approaches are good indicators of the currently intense effort to facilitate a natural interaction between human beings and robots.

Author Keywords

HRI, Human-Robot Interaction, NPL, natural language processing, natural language understanding, NLU, shallow semantic parsing, robotics

INTRODUCTION

Interactions in natural language dialogues are an essential part of human social exchanges: greetings, task-based dialogues for coordinating activities, topic-based discussions... The ability of future social and service robots to interact with humans in natural ways will depend mainly on developing capabilities of humanlike dialogue-based natural language processing (NLP) in robotic architectures. However, natural language processing on robots has at least the following six properties: real-time, parallel, spoken, embodied, situated, and dialogue-based. *Real-time* means that all processing must occur within the time frame of human processing, both at the level of comprehension as well as production. *Parallel* means that all stages of language processing must operate concurrently to mutually constrain possible meaning interpretations and to allow for the generation of responses -such as acknowledgements- while an ongoing utterance is being processed. *Spoken* means that language processing necessarily operates on imperfect acoustic signals with varying quality that depends on the speaker and the background noise. In addition to handling prosodic variations, this includes typical features of spontaneous speech such as various types of disfluencies, slips of the tongue, or other types of errors that are usually not found in written texts. *Embodied* means that robots have to be able to process multimodal linguistic cues such as deictic terms accompanied by bodily movements, or other gestures that constrain possible interpretations of linguistic expressions. It also means that the robot will have to be able to produce similar gestures that are expected by human interlocutors to accompany certain linguistic constructs. *Situated* means that, because speaker and listener are located in an

environment, they will have a unique perspective from which they perceive and experience events, which, in turn, has an impact on how sentences are constructed and interpreted. *Dialogue-based* means that information flow is not unidirectional but includes bidirectional exchanges between interlocutors based on different dialogue schemes. Whilst these six aspects present significant challenges there are also several advantages to natural language processing on robots that other NLP contexts do not have. For example, spoken natural language exchanges typically consist of shorter sentences with usually simpler grammatical constructions compared to written language -thus making parsing easier and more efficient. Moreover, the employed vocabulary is much smaller and the distribution of sentence types is different -including more commands and acknowledgements, and few declarative sentences compared to written language. Also, different from written texts, perceptual context can be used to disambiguate expressions, and most importantly, ambiguities or misunderstandings in general can often be resolved through subsequent clarifying dialogue.

1ST PERSPECTIVE: DESIGNING FLEXIBLE DIALOGUE-BASED ROBOTIC SYSTEMS, FROM DATA COLLECTED IN HUMAN INTERACTION EXPERIMENTS, IN THE CONTEXT OF A SEARCH TASK

It is difficult to anticipate a priori the wide variety of spoken task-based natural language interactions that will be initiated by humans during interactions with robots, even when the task is well defined. Hence, the researchers Scheutz, Cantrell, and Schermerhorn[1] conducted several human-human studies in a task where two humans had to coordinate their activities through remote audio communication only. Their transcribed interactions formed the Cooperative Remote Search Task (CReST) corpus, which the researchers used to identify natural interaction patterns as well as potential pitfalls.

Firstly it must be stated dialogue-based natural language interactions are quite different from natural language instructions, as they would be given in written form. For example, let's consider a written instruction for following a particular route such as "*Continue to walk straight, going through one door until you come to an intersection just past a whiteboard.*" An interactive version of the same instruction is substantially more complex:

Instructor: OK, continue to walk straight.

Robot (continuing straight): OK.

Instructor: You should be seeing a door in front of you.

Robot (looking out for a door): Yes. Instructor: Good, go through that door.

Robot (moving through the door): OK, I'm through the door.

Instructor: Alright. Keep going. There should be a whiteboard.

Robot (looking for whiteboard): OK, I'm not seeing it yet. Ah, there it is.

Instructor: Great, then you should see an intersection, go there.

Robot (looking out for an intersection while moving): Got it, OK.

It can be noted that natural language instructions are given piecemeal with the expectation of rapid feedback, and meanings as well as goals are negotiated through a sequence of dialogue moves and actions, rather than being fixed in a sequence of instructions. As a result, perception, natural language understanding (NLU), and behavior have to be tightly intertwined. Humans provide minimal information, happy to refine it or add to it during the ensuing dialogue interaction. Among the most frequently occurring issues identified in the CReST corpus are ungrammatical sentences, incomplete referential phrases, missing verbs, underspecified directions, frequent *ums*, *uhs*, and other disfluencies and pauses indicating cognitive load, and frequent coordinating confirmations and acknowledgments as dialogue moves including prosodically different *okays*, *yeahs*, and others.

Tackling Task-Based Dialogue HRI

Despite their seemingly very limited nature -based on vocabulary, the dialogues present a major challenge for HRI. It is clear that meanings are not constructed from sentences alone but from interactions that serve particular purposes and accomplish particular goals. Perception, action, and language processing in humans are obviously all intertwined, involving complex patterns of actions, utterances, and responses, where meaningful linguistic fragments result from their context together with prosodic, temporal, task and goal information, and not sentence boundaries. Consequently, new models of interactive natural language processing and understanding for HRI must be developed. In order to achieve human-level performance on robots, the timing of utterances, back-channel feedback, perceivable context -such as objects, gestures, eye gaze of the participants, posture, and others, as well as background and discourse knowledge, task and goal structures must be integrated. All of this poses both functional challenges and architectural challenges.

Functional challenges include firstly mechanisms for providing appropriate feedback that humans expect even while an utterance is still going on, using different kinds of acknowledgment based on dialogue moves; secondly, new algorithms for anaphora and reference resolution -one of the biggest problems in NLP as exposed by Levesque[2], using perceptual information as well as task and goal context; and thirdly mechanisms for handling various kinds of disfluencies and incomplete and ungrammatical utterances, including robust speech recognition, parsing, and semantic analysis.

Architectural challenges include firstly real-time processing of all natural language interactions within a human-acceptable response time -for example, typically acknowledgments have to occur within a few hundred milliseconds after a request; secondly integration of various natural language processing components -including speech recognition, parsing, semantic and pragmatic analyses, and dialogue moves- that allows for parallel execution; and thirdly automatic tracking of dialogue states and goal progress to be able to provide meaningful feedback and generate appropriate goal-oriented dialogue moves.

Handling Disfluencies

As afore-mentioned there are several basic types of disfluencies in the CReST corpus, amongst others: repetitions, insertions, abandoned utterances, and repairs.

Repetitions of exact words or word sequences:

Director: so two doorways and then you'll you'll be staring straight at a platform

Or several words in length:

Director: is that a new is that a new green box that you didn't tell me about

Insertions may be words (lexical) or nonwords (nonlexical). Nonlexical insertions include, for example, *uh*, *um*. Lexical insertions can be similar to repetitions but are not exact:

Director: how many box how many blue boxes do we have

Repairs, a subclass of insertions, denote instances in which one word is replaced by another. A simple one-word correction replacing *at* with *by*:

Searcher: one green box at the corner at by the end of the hallway

Abandoned utterances may or may not be followed immediately by a distinct sentence. In this example, the speaker abandons an utterance and does not start a new one:

Director: so pink boxes should uh only be in.

Rule-based algorithms for each of these disfluency types have shown some success. However, these methods are typically targeted at offline NLP, and rely on transcription features that are not usually provided by real-time speech recognizers. The researchers' method, given a possibly disfluent

utterance, is to produce only a partial parse with whatever can be made sense of. Having been trained on correct -that is, nondisfluent utterances, the parser, when faced with disfluent utterances, may do any of several things in order to discard such disfluences:

- **Attach Extra Nodes to the Root**:
  Each root node is viewed as a separate phrase; attaching disfluent words to the root creates semantically incomplete phrases that are subsequently discarded. In *to go to go into a room*, the disfluent initial nodes *to go* are both left unconnected to the rest of the graph, leaving only the relevant parts of the utterance -see Figure 1a.

- **Attach Extra Nodes to a Nearby Node:**
  Given the connection rules, this will often result in correct semantic output. For example, *you are um at a closed um door* should result in at(*listener,x*) where *x* is known to be a door that is closed. Figure 1b shows how this works for the first occurrence of *um*. The parser connects *at* as a pred child of *um*, which in turn is listed as a pred child of *are*. The definition of *are* indicates that it should take a predicate and a subject, and attach the subject to the predicate. The system creates an empty definition for *um* that allows it to attach the subject it is handed by *are* to its own predicate child *at*, resulting in the correct definition.

- **Leave Extra Nodes Unattached:**
  In Figure 1b the second um, between *a* and *closed*, is simply not attached to any node and is thus discarded.

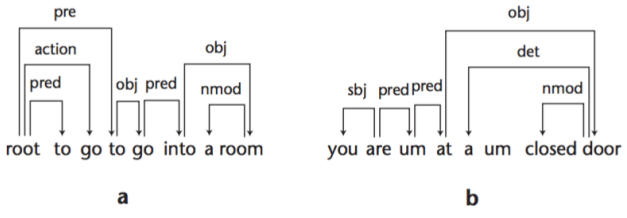

Figure 1 . Parsing disfluencies

There is still a long road ahead with many obstacles that need to be overcome before robots will be able to engage in humanlike natural language dialogues. However and as it has just been exposed, by starting with limited domains -such as instruction tasks, it is possible to make progress toward a not-too-distant future point in which the resultant architecture will be ready for transition into real-world application domains.

2ND PERSPECTIVE: TEACHING ROBOTS NEW HIGH-LEVEL ACTIONS THROUGH NL INSTRUCTIONS

A new generation of robots have emerged in recent years which serve as humans' assistants and companions. However, natural language based communication between them is difficult. Firstly, the robot's representation of its perceptions and actions are continuous and numerical in nature, but human language is discrete and symbolic. Secondly, the robot may not have complete knowledge about the shared environment and the joint task. Thus it is important for the robot to continuously acquire new knowledge through interaction with humans and the environment. A group of researchers of the Michigan State University[3] have developed certain techniques that allow humans to teach new high-level actions to the robot through natural language instructions. For example, let's suppose the robot does not understand the action *grab* when asked to perform "*Grab the blue block*". The human can teach the robot by step-by-step instructions: for instance, the human may specify "open gripper, move to the blue block, and close gripper". At the end of the teaching phase, the robot should capture what the *grab* action actually entails and more importantly it should be able to apply this action under different situations.

One important issue in action learning through natural language instructions is the representation of the acquired action in the robot's knowledge base so that it can be effectively applied to novel situations. To address this issue, a framework for action representation that consists of primitive actions and high-level actions was developed by the researchers: *primitive actions* capture both pre-conditions and effects which are directly linked to lower-level control functions; *high-level actions* -e.g., the action of *grab*, *stack*, etc.- are modeled by the desired goal states of the environment as a result of these actions.

Instead of directly modeling a high-level action as a sequence of actions/steps specified by the human, capturing the desired goal states provides much flexibility to address novel situations.

System Overview

In the investigation a SCHUNK arm manipulator and several blocks are used, placed on a table, which can be manipulated by the robotic arm.

When a human says "stack the blue block on the red block to your right". This utterance will first go through a semantic processor and the key semantic information from the utterance will be extracted and represented. Besides interpreting the language input, the robot continuously will perceive the shared environment using its camera and will represent the perceived world as a vision graph. The robot will also assess its own internal state such as the status of the gripper, whether it will be open or closed and where the gripper will be. Given the semantic information from the utterance and current vision graph, the robot will apply a reference resolver to ground referring expressions in the utterance to the objects in the physical world.

Since the robot doesn't understand how to perform the action *stack*, the human will then specify a sequence of actions/steps for the robot to follow, for example, "open gripper, move to the blue block, close gripper, move to the red block on your right, open gripper". The actions known by the robot are captured in the *action knowledge base* which specifies the desired goal state for each action. The *discrete planner* will come up with a sequence of operators which are directly connected to the continuous planner to perform the lower-level arm movements. By following the step-by-step instructions, the robot will come to a final state when the actions are completed. This final state will be captured and connected with the original action *stack* to serve as the representation for the action *stack(x,y)*. The new knowledge about *stack* is thus acquired and stored in the action knowledge base for future use.

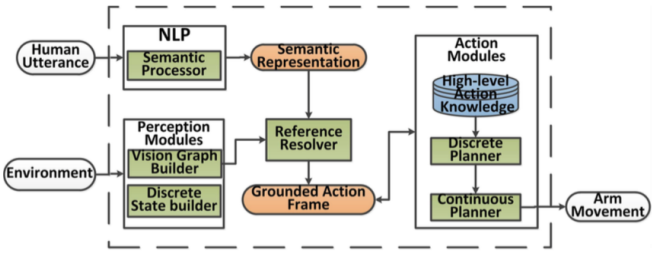

Figure 2. System Architecture

Natural Language Processing and Referential Grounding

The language understanding process consists of two sub-procedures: Natural Language Processing (NLP) and Referential Grounding.

1. Natural Language Processing: The NLP module is mainly composed by a semantic processor, which is used to extract action related information from utterances, implemented as a set of Combinatory Categorial Grammar (CCG) lexicons and a semantic parser.

2. Referential Grounding: The semantic representation by itself is still not understandable by the robot. It must be grounded to the robot's representation of perception and actions. The robot's perceived world is represented as a vision graph, which captures objects and their properties -in the numerical form- recognized by the computer vision algorithm. The linguistic entities and their relations -captured in the semantic representation- are represented as a language graph. Given these graph-based representations, referential grounding -i.e., Reference Resolver- becomes a graph matching problem that matches the language graph to the vision graph. Once references are grounded, the semantic representation of an action becomes a Grounded Action Frame.

Action Learning Through Language Instructions

A.- Action Knowledge Representation

The Action Knowledge Base represents action knowledge in a three level structure:

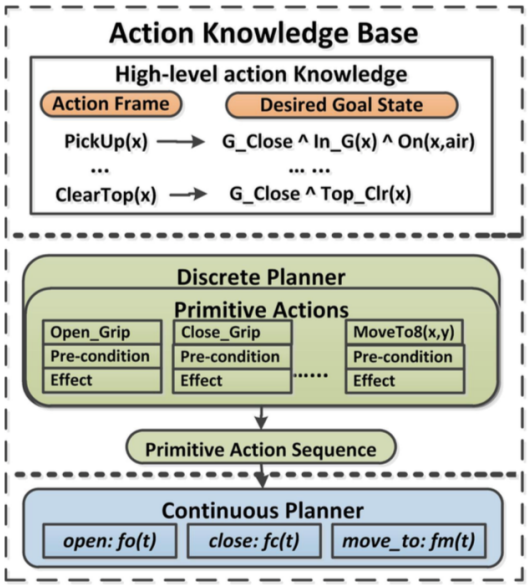

Figure 3. The three-tier action knowledge representation.

At the top level is the *High-level action knowledge*, which stores mappings from high-level action frames captured in human sentences to desired goal states. The middle level is a *Discrete Planner*, like STRIPS planner, maintaining domain information and operators. The bottom level is a *Continuous Planner* which calculates the trajectory to realize each of the atomic actions performed by the robotic arm.

1) Continuous Planner:

It is directly connected with the robotic control system and categorizes the basic operations the robot can perform.

For the SCHUNK arm used, the basic capability can be categorized as three atomic operations: open (i.e., open gripper), close (i.e., close gripper) and move to (i.e., move to certain location). Each of these three actions corresponds to a pre-defined end-effector trajectory parameterized by time t. Specifically, the open/close is a function $fo(t)/fc(t)$, which only controls the distance between two gripper fingers attached to the arm. For the move_to, given a destination coordinate, function $fm(t)$ calculates the trajectory for the end-effector to reach the location but does not change the distance of the gripper fingers. These functions are further translated to control commands through inverse-kinematics and sent to each motor to track the trajectory.

2) Discrete Planner:

It is used to automatically generate a plan -e.g., an action sequence- to achieve the goals specified by the high-level actions. The Discrete Planner is implemented as a STRIPS planner.

In the STRIPS, a domain is defined as a 2-tuple $D = \langle P,O \rangle$, where P is a finite set of predicates to symbolically represent the world and O is a set of legal operations distinguishing changes of the domain world. A planning problem is defined as another 2-tuple $Q = \langle I,G \rangle$, where I represents the world status at the beginning of the problem and G is the desired goal state if the problem can be solved.

In this specific domain the P set captures two dimensions of the domain:

- Robotic arm states: *G_Open/Close* stands for whether the gripper is open or closed, *G_Full/Empty* represents whether the gripper has an object in it; and *G_At(x)* describes where the arm locates.

- Object states: *Top_Uclr/Clr(x)* means whether the block x has another block on its top, *In/Out_G(x)* stands for whether it's within the gripper fingers or not; and *On(x,y)* means object x locates on y.

Each operator o, belonging to O, is defined as a set of pre-conditions and effects.

3) High-level Actions:

High-level actions are those specified by the human partners (e.g., *PickUp* and *ClearTop*). Currently, they are modeled by their desired goal states: when humans describe an action, they already have certain goal states in mind. And they want these goal states to be attained after robot performing the action.

B.- Action Acquisition

Action Acquisition means learning how to represent the human specified actions as high level action knowledge.

For a new unknown action to the robot, the human will typically go through a sequence of actions. A_i is the name of unknown action, and $c_1...c_k$ are the arguments which are objects in the scene. At the beginning of the instruction, the states of $c_1...c_k$ are $S_b(c_1...c_k)$. And at the end of the instruction another object states $S_e(c_1...c_k)$ are calculated. Then, the high level representation of $A_i(c_1...c_k)$ is calculated as:

$$\mathcal{A}_i(c_1...c_k) = (\mathcal{S}_e - \mathcal{S}_e \cap \mathcal{S}_b) \cup p_{grip}$$

where

$$p_{grip} \in \{G\_Open, G\_Close\}$$

depends on the arm status at the end of instruction.

As a conclusion and summing up, although there is a sound progress in this interaction way between humans and robots, the teaching/learning process is rather complex and many different kinds of exceptions could occur. For example, when teaching a new action, the robot could encounter another unknown action. Furthermore, besides actions, the robot may not perceive the environment perfectly. Therefore the human and the robot will need to maintain a common ground so that teaching/learning is successful.

3RD PERSPECTIVE: CREATING MAPPINGS FROM SENTENCES -USING SHALLOW SEMANTIC PARSING- TO EXPECTED ROBOT ACTIONS

Shallow semantic parsing, also called semantic-role-labeling, is the task of finding semantic roles for a given sentence. Semantic roles describe general relations between predicates and its arguments in a sentence. For example, in a sentence like "Mary gave the ball to Peter", "gave" is the predicate, "Mary" represents the semantic role *donor*, "the ball" represents the semantic role *theme*, and "Peter" represents the semantic role *recipient*.

The researchers of the Umeå University at Umeå in Sweden[4] propose a method by which the expected actions for a verbally uttered commands to a robot can be learned, such that the robot automatically can determine what to do when hearing a new sentence. The robot learns how to infer action and parameters from a set of labeled example sentences. Each sentence is parsed by a shallow semantic parser (Semafor system), which produces frames and associated semantic roles. If multiple frames occur, the frame related to the predicate is selected and denoted as the *primary* frame. Conditional probabilities for how these entities relate to expected actions and associated parameters are estimated and used to construct the necessary inference mechanisms. Furthermore each sentence is labeled with one of n_A robot actions a_1, ..., a_{n_A} and m_a associated parameters p_1 , ..., p_{m_a}

i	a_i	p_1	p_2	Expected function
1	BRING	object	recipient	Fetches object
2	TELL	message	recipient	Relays a message
3	COLLECT	object	source	Gathers objects
4	MOVE	location		Moves self to location
5	PUT	object	location	Places an object

Figure 4. Table of pre-programmed robot actions a_i with associated parameters p_1, p_2.

The proposed method comprises a learning part and an inference part.

The Learning Part

In the learning phase, each sentence in a training data set comprising N sentences is presented to the Semafor system, which outputs frames and associated semantic roles. If several frames are generated, the primary frame is selected.

The proposed method builds on the hypothesis that the expected action for a command can be inferred from the primary frame of the command. To initially test this hypothesis, statistics for combinations of primary frames and labeled actions for all sentences

are generated -see below Figure 5. The number of occurrences for each frame/action combination is shown, followed by the relative frequency after the / symbol. Most rows contain only one non-zero entry, thus supporting the hypothesis that the expected action can be inferred from the frame. However, some frames occur for more than one action, and many actions occur for several frames.

Frame \ Labeled Action	BRING	TELL	COLLECT	MOVE	PUT
1 Bringing	7/100%	0/0%	0/0%	0/0%	0/0%
2 Getting	4/100%	0/0%	0/0%	0/0%	0/0%
3 Giving	3/60%	2/40%	0/0%	0/0%	0/0%
4 Needing	2/100%	0/0%	0/0%	0/0%	0/0%
5 Desiring	4/100%	0/0%	0/0%	0/0%	0/0%
6 Telling	0/0%	8/100%	0/0%	0/0%	0/0%
7 Statement	0/0%	8/100%	0/0%	0/0%	0/0%
8 Political locales	0/0%	0/0%	0/0%	0/0%	0/0%
9 Being named	0/0%	0/0%	0/0%	0/0%	0/0%
10 Text	0/0%	1/100%	0/0%	0/0%	0/0%
11 Come together	0/0%	0/0%	4/100%	0/0%	0/0%
12 Amassing	0/0%	0/0%	4/100%	0/0%	0/0%
13 Gathering up	0/0%	0/0%	6/100%	0/0%	0/0%
14 Placing	0/0%	0/0%	2/11%	0/0%	17/89%
15 Motion	0/0%	0/0%	0/0%	14/82%	3/18%
16 Grant permission	0/0%	0/0%	0/0%	0/0%	0/0%
17 Departing	0/0%	0/0%	0/0%	1/100%	0/0%
18 Stimulus focus	0/0%	0/0%	0/0%	0/0%	0/0%
19 Have as requirement	0/0%	0/0%	0/0%	0/0%	2/100%
20 Locale by use	0/0%	0/0%	0/0%	0/0%	1/100%
21 Compliance	0/0%	0/0%	0/0%	0/0%	1/100%

Figure 5. Occurrences/frequencies for combinations of primary frames and labeled actions.

In order to infer expected action from the primary frame of a sentence, the conditional probability:

$$P(Action = a_i | Frame = f_j), \quad (1)$$

i.e. for the expected action to be a_i, given a primary frame f_j, is estimated. With simplified notation and by using the definition of conditional probability, (1) can be written as:

$$P(a_i|f_j) = P(a_i, f_j)/P(f_j), \quad (2)$$

which can be estimated from data by:

$$\widehat{P}(a_i, f_j) = \#(a_i, f_j)/N \quad (3)$$

and

$$\widehat{P}(f_j) = \#(f_j)/N, \quad (4)$$

where $\#(a_i, f_j)$ denotes the total number of sentences in the training data that are labeled with action a_i and for which Semafor determines f_j as primary frame. Hence, $P(a_i|f_j)$ can be estimated by:

$$\widehat{P}(a_i|f_j) = \#(a_i, f_j)/\#(f_j). \quad (5)$$

The n_F different frames that appear in the analyzed scenario have in total n_R distinct associated semantic roles with the following names: *Goal, Theme, Source, Recipient, Requirement, Cognizer, Event, Experiencer, Addressee, Message, Name, Text, Donor, Individuals, Mass theme, Path, Grantee, Action, Direction, and Dependent.* These semantic roles are in the following denoted $r_1, ..., r_{nR}$

Normally, each frame only has a few semantic roles defined. When parsing an input sentence s, Semafor assigns substrings of s as values to these semantic roles. The parameters for each robot action are related to specific semantic roles. A parameter p_i is regarded as matching (denoted by the symbol ~) a semantic role r_j, if p_i is a nonempty substring of the value of r_j:

$$p_i \sim r_j \equiv p_i \text{ is a nonempty substring of the value of } r_j. \quad (6)$$

Example: Let's assume that the sentence "Give me the glass" is labeled with action a_1 (i.e. BRING) and parameter p_1 = "glass". Semafor generates a primary frame f_3 (i.e. GIVING), and semantic role r_2 (i.e. Theme) is assigned the value "the glass" for the sentence. Hence, $p_1 \sim r_2$.

Another example of the capacity of the system: let's construct a classifier to infer expected action a_E for a sentence with a primary frame name f_E. To infer parameters for a_E, it must be estimated the probability that a parameter p_i for a_E matches a semantic role r_j, given that the primary frame is f_E. This can be written as:

$$P(p_i \sim r_j | f_E) = P(p_i \sim r_j, f_E)/P(f_E). \quad (7)$$

The probabilities on the right-hand-side of (7) can be estimated as follows:

$$\widehat{P}(p_i \sim r_j, f_E) = \#(f_E, p_i \sim r_j)/N \tag{8}$$

and

$$\widehat{P}(f) = \#(f_E)/N \tag{9}$$

where $\#(f_E, p_i \sim r_j)$ denotes the total number of sentences in the training data for which Semafor determines a primary frame f_E and a semantic role r_j, and the sentence is labeled with parameter p_i, satisfying $p_i \sim r_j$. The entity $\#(f_E)$ is the total number of sentences in the training data for which Semafor determines a primary frame f_E. Combining (7–9), it yields the following estimation:

$$\widehat{P}(p_i \sim r_j | f_E) = \#(f_E, p_i \sim r_j)/\#(f_E). \tag{10}$$

The Inference Part

A Bayes classifier is used to infer the expected action a_E for a sentence with a primary frame name f_E and semantic roles r_i, $i = 1, \ldots, n_R$. It works by inferring the action with highest conditional probability, as given by (1–5):

$$\begin{aligned} a_E &= \arg \max_{1 \le i \le n_A} \widehat{P}(Action = a_i | Frame = f_E) \\ &= \arg \max_{1 \le i \le n_A} \#(a_i, f_E)/\#(f_E) \\ &= \arg \max_{1 \le i \le n_A} \#(a_i, f_E). \end{aligned} \tag{11}$$

Each one of the parameters p_i^E, $i = 1, \ldots, m_{aE}$ required by action a_E is assigned the value of one of the semantic roles r_i, $i = 1, \ldots, n_R$ for the sentence. The procedure for inference of parameters follows the same principles as for inference of action in (7–10), and parameter values are assigned as follows:

$$p_i^E = r_{opt}, \tag{12}$$

where

$$\begin{aligned} opt &= \arg \max_{1 \le j \le n_R} \widehat{P}(p_i \sim r_j | f_E) \\ &= \arg \max_{1 \le j \le n_R} \#(f_E, p_i \sim r_j)/\#(f_E) \\ &= \arg \max_{1 \le j \le n_R} \#(f_E, p_i \sim r_j). \end{aligned} \tag{13}$$

As a conclusion, it must be mentioned that through the proposed method the researchers' hypothesis was valid for more than 88% of the tested sentences. Expected actions and parameters were correctly inferred for 68% of the cases. Given the large variety of sentences, and the small data set being used, the result is considered both surprising and promising. Better results can be expected by adding more data.

	Sentence	Expected action	p_1	p_2
1	move the chairs to the kitchen	PUT	chairs	the kitchen
2	Move 2 meters to the left	MOVE	2 meters	to the left
3	I want a glass of water.	BRING	a glass of water	
4	Robot, tell Ola the name of the book.	TELL	the name of the book	Ola
5	stash the balls in the wardrobe.	PUT	the balls	in the wardrobe
6	package all glasses into nice parcels.	PUT	all glasses	into nice parcels
7	Gather all the green balls.	COLLECT	all the green balls	
8	Robot, tell Ola the color of the ball.	TELL	the color of the ball	Ola
9	Gather dust in the room.	COLLECT	dust	in the room
10	Go to the tire storage.	MOVE	the tire storage	
11	Robot, tell the direction of the exit to me.	TELL	the direction of the exit	me
12	Bring Ola's book to me.	BRING	Ola's book	me

Figure 6. Examples of sentences used for training and evaluation. Each sentence is labeled with expected action and associated parameter(s).

	Primary frame	Semantic role/value	Semantic role/value	Semantic role/value
1	MOTION	Theme/the chairs	Goal/to the kitchen	
2	MOTION	Theme/2 meters	Goal/to the left	
3	DESIRING	Experiencer/I	Event/a glass of water	
4	TELLING	Speaker/Robot	Addresse/Ola	Message/the name of the book
5	PLACING	Theme/the balls	Goal/in the wardrobe	
6	PLACING	Theme/all glasses	Goal/into nice parcels	
7	COME TOGETHER	Individuals/all the green balls		
8	TELLING	Speaker/Robot	Addresses/Ola	Message/the color of the ball
9	COME TOGETHER			
10	MOTION	Goal/to the tire storage		
11	TELLING	Speaker/Robot	Addresses/to me	Message/the direction of the exit
12	BRINGING	Theme/Ola's book	Goal/to me	

Figure 7. Semantic parses of the sentences in the table of Figure 6, as given by the Semafor system. The table shows primary frame name and some of the generated semantic roles for the frame.

CONCLUSION

In this paper the use of natural language processing techniques applied to robot-human being interaction has been analyzed from three different and, from my standpoint, complementary perspectives: Firstly, the design of flexible dialogue-based robotic systems,

from data collected in human interaction experiments, in the context of a search task; secondly, techniques that allow humans to teach new high-level actions to robots through step-by-step natural language instructions; and thirdly, a method to create mappings from sentences -using shallow semantic parsing- to expected robot actions.

As it can be realized AI researchers are striving to ease the interaction between humans and robots through natural language processing. Nonetheless, beyond the linguistic issues, there are other important open questions related to a robot's physical appearance and capabilities. It must, for example, be tested whether people will even seriously engage in humanlike conversations with robots that have very different physical forms -for example, wheels, no heads, and others, and very different physical capabilities. In any case the future and natural inclusion of robots in our daily routines will depend mainly on their NLP abilities; thus it seems the three afore-mentioned approaches are guiding us to the right direction.

REFERENCES

1. Matthias Scheutz, Rehj Cantrell, Paul Schermerhorn, *Toward Humanlike Task-Based Dialogue Processing for Human Robot Interaction.* AI Magazine, Winter 2011.

2. Hector J. Levesque, *The Winograd Schema Challenge.* Department of Computer Science. University of Toronto, Canada. 2011.

3. Lanbo She, Yu Cheng, Joyce Y. Chai, Yunyi Jia, Shaohua Yang, and Ning Xi, *Teaching Robots New Actions through Natural Language Instructions.* The 23rd IEEE International Symposium on Robot and Human Interactive Communication. August 25-29, 2014. Edinburgh, Scotland.

4. A. Sutherland, S. Bensch, and T. Hellström, *Inferring Robot Actions from Verbal Commands Using Shallow Semantic Parsing.* Department of Computing Science, Umeå University, Umeå, Sweden. International Conference on Artificial Intelligence, ICAI'15.