ECAI 2014
T. Schaub et al. (Eds.)
© 2014 The Authors and IOS Press.
This article is published online with Open Access by IOS Press and distributed under the terms
of the Creative Commons Attribution Non-Commercial License.
doi:10.3233/978-1-61499-419-0-57

Effective and Robust Natural Language Understanding for Human-Robot Interaction

Emanuele Bastianelli(‡), **Giuseppe Castellucci**(•), **Danilo Croce**(†), **Roberto Basili**(†), **Daniele Nardi**(⋆)
(†) DII, (‡) DICII, (•) DIE - University of Rome Tor Vergata - Rome, Italy
{bastianelli,castellucci}@ing.uniroma2.it, {basili,croce}@info.uniroma2.it
(⋆) DIAG - Sapienza University of Rome - Rome, Italy
nardi@dis.uniroma1.it

Abstract. Robots are slowly becoming part of everyday life, as they are being marketed for commercial applications (viz. telepresence, cleaning or entertainment). Thus, the ability to interact with non-expert users is becoming a key requirement. Even if user utterances can be efficiently recognized and transcribed by Automatic Speech Recognition systems, several issues arise in translating them into suitable robotic actions. In this paper, we will discuss both approaches providing two existing Natural Language Understanding workflows for Human Robot Interaction. First, we discuss a grammar based approach: it is based on grammars thus recognizing a restricted set of commands. Then, a data driven approach, based on a free-from speech recognizer and a statistical semantic parser, is discussed. The main advantages of both approaches are discussed, also from an engineering perspective, i.e. considering the effort of realizing HRI systems, as well as their reusability and robustness. An empirical evaluation of the proposed approaches is carried out on several datasets, in order to understand performances and identify possible improvements towards the design of NLP components in HRI.

1 Natural Language Processing and Human Robot Interaction

Human Robot Interaction (HRI) is a novel research field aiming at providing robots with the ability of interacting in the most similar way the humans do. Such ability has become a key issue since robots are nowadays attracting the interesting of commercial and public applications, e.g. viz. telepresence, cleaning or entertainment. In this context, Natural Language HRI studies the interaction between humans and robots focusing on natural language. Ideally, robots should be able to solve human language references to the real world application contexts (e.g. find the place of the *can* in a map against the phrase *bring the can in the trash bin*) or in abstract dimensions (e.g. solving anaphoric references).

Natural Language Processing techniques can be heavily applied in this context. Even if Automatic Speech Recognition (ASR) systems can be used to recognize user utterances, several issues arise in obtaining the suitable mapping between language and the relative robotic actions. First, we need to *capture* the intended meaning of the utterance, and then map it into robot-specific commands. This task, that is filling the gap between the robot world representation and the linguistic information, is a typical form of semantic parsing. Semantics is the crucial support for grounding linguistic expressions into objects, as they are represented in the robot set of beliefs (i.e. robot knowledge). Moreover, in HRI, different scenarios need to be considered. In some situations, e.g. rescue robotic tasks, the precision is crucial and no command misunderstanding is allowed.

Recently, works in the interpretation of natural language instructions for robots in controlled environments have been focused on specific subsets of the human language. The interpretation process is mainly carried out through the adoption of specific grammars that describe the human language allowed. It gives, for example, a robotic platform the capability to speak about routes and navigation across maps [18, 7]. Grammar based approaches lead often to very powerful and formalized systems that are, at the same time, very constrained, and focused onto a limited domain. However, the development of wide coverage grammars require skilled profiles and may be very expensive.

On the opposite, in more complex and less restricted scenarios, such as house serving tasks, people do not follow a-priori known subsets of linguistic expressions. This requires robust command understanding processes, in order to face the flexibility of natural language and the use of more general approaches, able to adjust the language models through observations. In many Natural Language Processing tasks, where robustness and domain adaptation are crucial, e.g. Question Answering as discussed in [13], methods based on Statistical Learning (SL) theory have been successfully applied. They allow to cover more natural language phenomena with respect to rule based approaches. This paradigm has been applied in the HRI field by researches with different background, e.g. Robotics or NLP [9, 16, 21]. Language learning systems usually generalize linguistic observations, i.e. annotated examples, into rules and patterns that are statistical models of higher level semantic inferences. As a result, these approaches aim at shifting the attention from the explicit definition of the system behavior, e.g. the definition of large scale knowledge bases, to the characterization of the expected behavior of the system through the manual development of realistic examples. This is very appealing from an engineering perspective as complex systems can be specialized by people that are only expert of the targeted domain. In this paper, we will discuss both approaches providing two existing Natural Language Understanding workflows for HRI, and attempting to answer to the following research questions:

- how are the performances of the two approaches affected by the open scenarios that are typical of HRI in service robotics? how are they adaptive to the language and how are they robust to the noise in speech acquisition?
- how do the approaches compare in terms of ease of implementa-

tion and extensibility?

- which improvements should be addressed in order to obtain better performances?

We investigate the above issues by discussing two different architecture for HRI command parsing. First, we discuss a grammar based approach: it is based on grammars developed in the context of the Speaky For Robots[1] [8] project. Then, a modular approach, based on a free-from speech recognizer and a statistical semantic parser presented in [10], is discussed. An empirical evaluation of the proposed solutions is carried out on several datasets, each characterizing a different scenario as well as command complexity. First, we will measure the quality of the command recognition capability of both architectures, to evaluate the respective robustness. Then the modular workflow will be evaluated by using general purpose and publicly available resource, as well as by extending the training material with ad-hoc examples. Finally, the modular processing chain will be unfolded in order to measure the specific sources of error.

In the rest of the paper, Section 2 presents previous works about NL for HRI. Section 3 discusses the proposed framework. In Section 4 the experimental evaluation is presented, while Section 5 discusses the possible future directions of this work.

2 Existing NLU approaches for HRI

The ability of executing a command given by a user depends directly on the robot capability of understanding the human language. A sequence of non-trivial steps are required to obtain the actual action to perform. The audio signals of user utterances must first be transcribed into text; then this is analyzed at different linguistic levels.

Regarding the Speech Processing stage, grammar based approaches, as discussed in [6, 7, 18] can be applied. The grammars define the controlled language handled by the system (i.e. all the recognizable sentences). With a grammar-based specification of the language, it is possible to attach semantic actions to individual rules and the semantic interpretation of the target commands is built via the composition of the semantic actions, as in [6]. Grammar-based Speech Recognition has been adopted also in [4] in the context of a *Human-Augmented Mapping* task. In this work, recognition grammars are augmented with semantic attachments, enabling the composition of a linguistically motivated representation of the commands expressed as output of the recognizer. Where the speech transcription is already provided, grammars have been also used to parse the admissible sentences, as in [12]. Here, a general POS-tagger is applied on the input string; then, the grammar is used to identify the proper POS sequences, defining the subset of possible recognized commands according to the tagging. Grammars are the traditional NLP approaches to text analysis aimed at driving the semantic interpretations of user utterances. NLP approaches based on formal languages have thus been widely employed in many HRI systems, such as the semantic parsing proposed in [7], where a meaning representation based on *Discourse Representation Structures* [11] is obtained directly from the speech recognition phase. Similarly, in [18], *Combinatory Categorial Grammar* (CCG) are used to produce a representation in terms of *Hybrid Logics Dependency Semantics* [17] logic form.

Grammar based approaches usually guarantee good performances (especially high precision) over the natural language fragment they have been designed for. However, general HRI systems tend to face a wider range of linguistic phenomena, and need to exhibit high flexibility and robustness. In the last decade, Statistical Learning (SL) techniques have been successfully applied to NLP, and a set of general purpose robust technologies have been developed, as discussed in [19]. Free-form speech recognition engines, syntactic as well as semantic parsers, based on different SL approaches are today available. Moreover, their application in different NL processing chains for complex tasks makes them suitable for application also in HRI. The work by Chen and Mooney [9] is an example of this approach, dealing with a simulated robotic system able to follow route instructions in a virtual environment. A NLP processing chain is built, that learns how to map commands in the corresponding verb-argument structure using String Kernel-based SVMs. The work in [23] proposes a system that learns to follow NL-directions for navigation, by apprenticeship from routes through a map paired with English descriptions. A reinforcement learning algorithm is applied to determine what portions of the language describe which aspects of the route. In [16] the problem of executing natural language directions is formalized through *Spatial Description Clauses* (SDCs), a structure that can hold the result of the spatial semantic parsing in terms of spatial roles. The SDCs are extracted from the text using *Conditional Random Fields* that exploit grammatical information obtained after a POS-tagging process. The same representation is employed in [21], where the probabilistic graphical model theory is applied to parse each instruction into a structure called *Generalized Grounding Graph* (G^3). Here SDCs are the base component of the more general structure of the G^3, representing the semantic and syntactic information of the spoken sentence.

In order to address the research question arising in the development of a spoken language interface in the context of HRI, in the paper, we evaluate two independent processing frameworks: one dedicated grammar-based system and a second, general approach, based on a cascade of existing and data-driven NLP components. The two systems use different technologies for speech recognition and linguistic analysis, while sharing the same semantic representation for encoding user commands, inspired by Fillmore's *Frame semantics* [14]. This enables us to investigate the reuse of available and domain-independent resources for training, such as FrameNet [2]. Accuracy of the resulting interpretation process is thus studied across a wide range of phenomena in HRI, whose coverage is critical and may be improved by general purpose (off-the-shelf) NLP tools.

3 Understanding NL Commands

In this section, design of our processing frameworks related to the target robotic platform and the application scenario are first presented, then a detailed description of the two systems and the design of their specific workflows is provided. Existing methodological results and tools can be used to acquire useful semantic representations to understand robot commands. Let us consider a house environment where a robot receives vocal instructions. First, such a command can be expressed just in agreement with existing linguistic and cognitive theories. Moreover, aspects of the obtained representation can be further specified to better reflect the targeted human instructions of the robotic environment. In this work, we use *Frames Semantics* [14] as a meaning representation language for the different commands. It generalizes the actions or, more generally, the *agent experiences* into *Semantic Frames*. Each frame corresponds to a (physical) situation (e.g. a general concept or an action/event), and it is fully expressed in terms of its participating entities, i.e. its *semantic arguments*. Arguments play specific roles with respect to the situation described by the frame, so that a full (i.e. understood) situation requires a core set of arguments to be all recognized.

[1] `http://labrococo.dis.uniroma1.it/?q=s4r`

The reference to general theories is also interesting in an empirical perspective: large scale resources are usually associated with them (e.g. annotated corpora) so that a cost-effective acquisition of data-driven models is enabled. In the FrameNet corpus [2] lexical entries (e.g. verbs, nouns and adjectives) are linked to *Frames*, and the arguments, expressing the participants in the underlying event, are defined by different *Frame Elements* (i.e. the semantic arguments). FrameNet provides an extensive collection of frames associated to a large scale annotated corpus of 150k sentences. For example, according to the FrameNet paradigm, the sentence "*Bring the can in the trash bin*" is annotated as follows: $[\textbf{\textit{bring}}]_{Bringing}$ $[\textit{the can}]_{Theme}$ $[\textit{in the trash bin}]_{Goal}$ where BRINGING is the communicated event, and its frame elements THEME and GOAL are both detected and compiled into the input command for the robot.

Table 1. Semantic Frames investigated in a home environment

ATTACHING	CLOSURE	PERCEPTION_ACTIVE
BEING_IN_CATEGORY	ENTERING	PLACING
BEING_LOCATED	FOLLOWING	RELEASING
BRINGING	GIVING	SEARCHING
CHANGE_DIRECTION	INSPECTING	TAKING
CHANGE_OPERATIONAL_STATE	MOTION	

As the set of possible commands corresponds to what the targeted real robot can accomplish in the home environment [8], we can effectively proceed by mapping each command to a semantic frame, i.e. conveniently selected among the FrameNet frame system. This selected subset of FrameNet-inspired frames is shown in Table 1, where each frame corresponds to a set of robot commands. Through a static mapping from the frame semantics to the corresponding syntactic structure, also the syntax for every commands is made available.

The final target of the interpretation process of the utterance "*bring the can in the trash bin*" (whatever NLU framework is applied) is thus the semantic analysis and the extraction of the command:

$$\texttt{BRINGING}(\texttt{theme}:[the, can], \texttt{goal}:[in, the, trash, bin])$$

In the experimental scenario, we consider a wheeled robot capable of moving freely in a known environment (i.e. the related map is previously acquired). The two NLP chains both generate the above final command and submit it to the robot, that makes use of its knowledge about the allowed frame structures for the grounding and planning stages. The environment is represented in the robot as a *Semantic Map* (SM) [4] describing accessible rooms and objects. It also contains terminological facts defining spatial relations between objects. It is worth noting that the selected frames derives from the analysis made for the Speaky for Robots project, that defines a set of actions a general house service robot could perform. For this reason, since our robot is not provided with any gripper, we consider just the implied movement actions for those frames that foresee grasping or other capabilities. As a consequence, the *bringing* action will correspond for our robot into the motion from its actual position to the position of the (known) *trash bin* object of the scene. Accordingly, the selected commands may refer to actions that are complex or still ambiguous from a robotic point of view. The final grounding (e.g. computing the destination coordinates) is realized by interpreting the objects referred by the related semantic role (i.e. the GOAL role in most cases): a Prolog interpreter translates the command into a logical form and execute it against the KB. More details about the two workflows are hereafter provided.

3.1 A grammar-based workflow

The system based on speech recognition grammars is an extension of the one presented in [8] and lately improved in [4]. It relies on a speech engine whose grammar is extended according to the Frame Semantics.

From Voice to Semantic Interpretation. The first process is speech recognition whose module yields a parse tree that encodes both syntactic and semantic information based on FrameNet. The tree corresponds to the grammar rules activated during the recognition, augmented through post-processing by an instantiation of the corresponding semantic frame. The compilation of the suitable robot command proceeds by visiting the tree and mapping each recognized frame: this final command is then interpreted by the robot.

Jointly modeling the syntactic-semantic phenomena. The recognition grammar targets the syntactic and semantic phenomena that arise in the typical sentences of home HRI applications. In preliminary experimental stages, these sentences have been gathered by direct interviews to the robot users, i.e. people well aware of the set of the actions executable by the robot. The resulting grammar encodes a set of imperative and descriptive commands in a verb-arguments structure. Each verb is retained as it directly evokes a frame, and each (syntactic) verb argument corresponds to a semantic argument. The lexicon of arguments is semantically characterized, as words playing argument roles are constrained by one or more semantic types. For example, for the semantic argument THEME of BRINGING only the type *Transportable Object* is allowed so that a subset of words referring to things transportable for the robots (e.g. *can*, *mobile phone*, *bottle*) are accepted. Figure 1 shows a segment of the grammar defined for BRINGING.

Moreover, wildcards element are used (i.e. terminal symbols that enable the grammar to skip irrelevant segments). This increases the grammar flexibility against phenomena such "*could you please bring the can in the trash bin*" vs. "*bring the can in the trash bin*", from which the same frame is derived.

```
Bringing → Target Theme Goal | ...
Target → bring | carry | ...
Theme  → Transportable_objects
Transportable_objects → can | book | bottle | ...
```

Figure 1. A subset of the BRINGING grammar

The grammar allows the designer to fully control the supported language and to constrain the commands handled by the robot. However, it requires ad hoc rules, limiting the use of heterogeneous linguistic phenomena.

3.2 A general purpose and modular workflow

As shown in Figure 2, a general workflow can be devised from existing tools, such as a free-form ASR, generic morpho-syntactic parser and statistical semantic parser. In the rest of this section, a specific chain is discussed in terms of its three main modules.

Voice to Text Transcription. A free-form speech-to-text engine is employed, i.e. an engine mainly based on statistical analysis of very large corpora. We used the official Google speech APIs of an Android environment to obtain the transcriptions of an audio dataset.

Grammatical Analysis. The last two decades of NLP research have seen the proliferation of tools and resources that reached a significant maturity after 90's. We decided to perform a morpho-syntactic analysis with a State-of-the-Art natural language parser, i.e. the Stanford Core NLP platform[2]. It includes tokenization, POS tagging, Named

[2] http://nlp.stanford.edu/software/corenlp.shtml

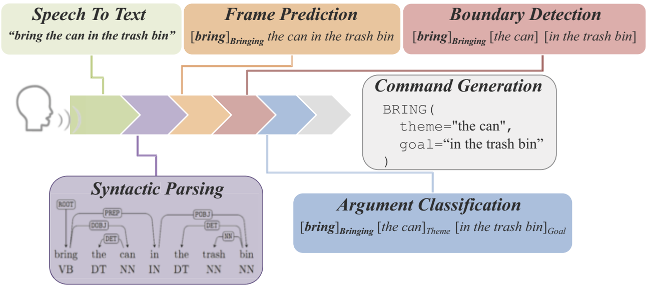
Speech To Text
"bring the can in the trash bin"
Frame Prediction
[bring]Bringing the can in the trash bin
Boundary Detection
[bring]Bringing [the can] [in the trash bin]
Command Generation
BRING(
theme="the can",
goal="in the trash bin"
)
Syntactic Parsing
ROOT
PREP
DOBJ
DET
POBJ
DET
NN
bring the can in the trash bin
VB DT NN IN DT NN NN
Argument Classification
[bring]Bringing [the can]Theme [in the trash bin]Goal

Figure 2. The free-form workflow

Entity Recognition as well as parsing and it is mostly based on statistical language processing models. Information extracted by the NL Parser, e.g. the grammatical category of a word (Part of Speech), are crucial for the later stages, in order to achieve a good command recognition accuracy.

Extracting commands through Semantic Arguments. Semantic Role Labeling is carried out through Babel, a general purpose system [10] based on FrameNet. The Semantic Parser realized by the Babel platform performs three main tasks: Frame Prediction (*FP*), Boundary Detection (*BD*) and Argument Classification (*AC*). Given a sentence s, *FP* is the task responsible of recognizing the type of the intended action, reflected by the event evoked by s. A multi-classification schema, based on the $SVM^{multiclass}$ approach by [15], is applied over the selected *ad-hoc* morpho-syntactic features, e.g. Part-of-Speech sequences. *BD* is the task of recognizing the spans of the arguments of an action, i.e. what we call its *semantic arguments*. From a computational perspective, we model this problem as the recognition of the *boundaries* of the arguments involved in a given sentence and frame. For example, with respect to "*bring the can in the trash bin*" and the frame BRINGING, the *BD* module should recognize the boundaries **bring** [*the can*] [*in the trash bin*] as two different arguments by identifying the start and the end of each chunk. Here, a different classification schema has been adopted, based on a Markovian formulation of a structured Support Vector Machine, i.e. SVM^{hmm} proposed in [1]. The SVM^{hmm} algorithm learns a model isomorphic to a k-order Hidden Markov Model. The *BD* phase is then modeled as a sequence labeling task over sentence observations, such as lexical properties (i.e. words) and syntactic properties (e.g. Part-of-Speech tags). The classifier associates a special tag to each word in the sentence, suggesting that it is a Begin (`B`), Internal (`I`) or Outer (`O`) token with respect to the argument boundaries: a correct labeling of the example sentence is represented as `O`-*bring* `B`-*the* `I`-*can* `B`-*in* `I`-*the* `I`-*trash* `I`-*bin*. Finally, the *AC* task aims at assigning a label to each recognized span from the *BD* phase, e.g. THEME to *the can* and *Goal* to *in the trash bin*. Again, the structured formulation of SVM^{hmm} is applied with different target classes, i.e. the role labels.

With respect to the previous grammar-based workflow, this solution requires labeled data for the final system development. However, the proposed workflow can be plugged in different scenarios, only adapting the training material: this activity can be accomplished also from people not aware of the system architecture. The labeled examples can be directly derived from already existing resources, e.g. the FrameNet corpus, or can be easily extended with ad-hoc material.

4 Experimental evaluation

An in depth evaluation of the two workflows has been carried out to address the main research questions posed in Section 1.

4.1 Experimental Setup

Three datasets representing different working conditions have been employed. Each dataset includes a set of audio recordings paired with the correct transcriptions. Utterances have been pronounced by different users, so that multiple audio versions of the same sentence are included. Table 2 reports the number of audio files, sentences and commands of the three datasets[3].

The **Grammar Generated** (GG) dataset refers to sentences generated by the grammar used by the grammar-based workflow of Section 3.1. Even if all sentences are grammatically and lexically covered by the grammar, speech imperfections and noise may significantly affect limit the quality of the transcription step.

The **Speaky 4 Robots Experiment** ($S4R$) dataset contains sentences pronounced by people in the context of the Speaky for Robots project experiments. Speakers were all aware about the lexicon handled by the grammar. They used free spoken English, including richer syntactic structures. These commands thus exhibit linguistic phenomena only partially covered by the grammar.

The **Robocup** (RC) dataset has been gathered during the Robocup@Home 2013 competition, presented in [24]. A Web portal describing home situations, enriched by images, has been used to suggest the home scenario: users were asked to input realistic robot commands. Expressions uttered here exhibit large flexibility in lexical choices and syntactic structure. This dataset is much more variable representing a realistic application for the two workflows.

The three corpora have been manually labeled with syntactic (Part-of-Speech tags and dependency trees) as well as semantic (i.e. Frames and Frame Elements) information, according to the set defined in Table 1. More details about the annotation process can be found in [5]. The grammar-based workflow follows the experimental setup discussed in [8]. About 1,67 different argument structures per frame are modeled in the grammar, through entries such as the ones shown in Figure 1. The free-form workflow consists in the chain including the Google Android ASR[4], the CoreNLP 3.3.0 for syntactic processing and Babel as the SRL system, configured as in [10].

Table 2. The evaluation annotated corpora

	#audio files	#sentences	#commands
GG	137	48	48
S4R	141	96	99
RC	292	177	195

In order to study the performance of the data-driven component of the free-form workflow (i.e. the SRL system), different experimental settings, i.e. including (or not) additional training material, have been employed. In the **only FrameNet** setting (FN), the SRL system is only trained on the labeled sentences from the FrameNet corpus and tested on the HRI corpus (see Table 2). Only examples of frames appearing in Table 1 and evoked by verbs have been selected, for a total of 5,162 FrameNet sentences, used for the parameter estimation through a 5-fold policy. In the **Hybrid** setting (H), 66% of annotated examples in each HRI corpus of Table 2 are added to the FrameNet material for training. In this way a 3-fold evaluation schema is enabled: cyclically, three different tests are made on the remaining 33% of each test corpus. The macro-average achieved over three measures is thus computed as the final performance. Notice that the Google API or the CoreNLP processing chain were not re-trained, as the former was out of our control, while re-training the syntactic parser was not considered effective, as confirmed by the data in Table 5.

[3] A single sentence may contains different commands provided in sequence.
[4] Transcription of the datasets has been carried out on February 24th, 2014

4.2 Evaluating the workflows

In order to address the question “*how are the performances of the two approaches affected by the open scenarios?*”, we evaluated the two workflows on the task of recognizing the complete robot command from an audio stream. We expect a robust system to provide good results across the different conditions, ranging from the restricted GG dataset, to the more complex $S4R$ and RC ones. We evaluated both systems on two variants of the task. First, we measure the *Action Recognition* (**AR**) capability, shown in Table 3, as the ability to detect just the command without its potential arguments. In the full *Command Recognition* (**CR**) measure, instead, the command as well as all its argument are to be exactly recognized, as reported in Table 3. This is a far more complex task, as individual words must be assigned to their correct argument and a single error in a word of an argument transcribed by the ASR may invalidate the recognized command. The performances are measured through the F1-Measure over sentences: to the harmonic mean between Precision, i.e. the percentage of commands correctly detected by the system[5], and the Recall, i.e. the percentage of commands correctly recognized in the dataset.

The grammar-based approach (the Grammar column) achieves best results on the *GG* (F1=0.78) in the **AC** and full **CR** (F1=0.42) tasks. Of course the grammar is accurate in sentences of its own design domain. However, when the syntactic complexity grows, e.g. in the $S4R$ dataset, performance drops of the F1 are observed (0.63 and 0.25). The drop is even more noticeable in RC, where richer lexical information was employed. The results are quite different for the free-form workflow (i.e. columns FN and H). When only FrameNet (i.e. general resources) is used, performances over the *GG* corpus decrease, i.e. 0.62 and 0.18. In $S4R$ results improve, especially in the **CR** task, as confirmed by the F1 scores in Table 3. The results exhibited by the FN setting are noticeable, as *only general purpose resources have been here used*. Moreover, when *ad-hoc* material is added in the training phase, performances still improve: the H setting achieves F1 of 0.53 and 0.18 on RC.

Table 3. Recognition performance starting from the audio file

	Action Recognition								
	Grammar			FN			H		
	P	R	F1	P	R	F1	P	R	F1
GG	0.80	0.77	0.78	0.86	0.48	0.62	0.86	0.48	0.62
S4R	0.69	0.57	0.63	0.86	0.47	0.61	0.86	0.47	0.61
RC	0.21	0.07	0.11	0.79	0.34	0.48	0.76	0.41	0.53

	Command Recognition								
	Grammar			FN			H		
	P	R	F1	P	R	F1	P	R	F1
GG	0.43	0.41	0.42	0.25	0.14	0.18	0.29	0.16	0.21
S4R	0.28	0.23	0.25	0.36	0.20	0.26	0.44	0.24	0.31
RC	0.04	0.02	0.02	0.25	0.11	0.15	0.26	0.14	0.18

In order to study sources of error in our chain, we measured the Word Error Rate (WER, as in [20]) of the speech transcription, as reported in Table 4. It confirms the robustness of the free-form approach against complexity. In the RC corpus, the WER of the grammar-based approach is 0.801 as most of words are wrongly transcribed. The WER of the Google API is 0.362. We also measured the accuracy of the POS tagging phase (see Table 5), that triggers Babel SRL. Accuracy of the assigned POS tags is very high for commands, in line with the results discussed in [22].

Major source of error is the Speech recognition that is amplified along the chain. Manual inspection of transcriptions revealed that many, apparently minimal, errors can compromise the entire chain. In the sentence “*bring the paper near the television*” transcribed as “*ring the paper near the television*”, the main verb is poorly recognized, i.e. *ring* vs. *bring*, so that no frame can be recognized.

Table 4. Speech Recognition Word Error Rate

	GramBased	Google
GG	0.165	0.176
S4R	0.434	0.379
RC	0.801	0.362

Table 5. CoreNLP Pos-Tagging Accuracy

	Accuracy
GG	0.963
S4R	0.946
RC	0.951

In order to avoid the bias introduced by the limited performance of the ASR stage, the semantic parsing chain is evaluated alone providing gold information as input to every module. Table 6 shows the results of the **AC** and the **CR** tasks of the free-form workflow. It is worth noticing that a similar evaluation of the grammar based workflow is not possible as in that case the action and command are computed jointly during the speech recognition phase. If compared with the previous settings, the F1 score are in general significantly higher. An interesting result in Table 6 is the fact that the F1 of the H setting is alway over 0.6 and stable across all corpora. It is an important finding that addresses the question “*how do the approaches compare in terms of ease of implementation and extensibility?*”. The free-form and trainable workflow seems much more portable across the working conditions, as shown by the F1 results on the $S4R$ and RC corpora: these are comparable or higher with respect to the grammar-based method, although the FN chain makes no use of domain specific resources. The performance drops of the grammar-based workflow outside the *GG* corpus may affects its applicability. Finally, optimization of the data driven modules allows the Hybrid settings to outperform the other workflows.

Table 6. Recognition performances from gold standard transcriptions

	Action Recognition					
	FN			H		
	P	R	F1	P	R	F1
GG	0.842	0.701	0.765	0.825	0.825	0.825
S4R	0.815	0.748	0.780	0.821	0.844	0.832
RC	0.838	0.629	0.719	0.926	0.912	0.919

	Full Command Recognition					
	FN			H		
	P	R	F1	P	R	F1
GG	0.325	0.270	0.295	0.613	0.613	0.613
S4R	0.496	0.456	0.475	0.642	0.660	0.651
RC	0.462	0.347	0.396	0.657	0.647	0.652

4.3 Measuring the error propagation

In order to address the question “*which improvements should be addressed in order to obtain better performances?*”, an in-depth analysis of the Babel system is reported. The sources of error within its free-form workflow are discussed. In particular, we are interested in analyzing the error propagation through the chain under different conditions. First of all, we need to verify how errors propagate across the chain. This is obtained by feeding each module both with gold standard input information and non-gold input. Again, we evaluated the method by using only FrameNet (FN) to train the system or by retraining with HRI material as in the Hybrid (H) setting.

In the Frame Prediction (FP) phase, F1 measures the system quality in correctly recognizing the frame(s), i.e. the robotic action, of

[5] Notice that both the grammar and the free-form workflow may provide null answers, i.e. no labeling, in several situations, e.g. when the quality of the transcription is too low.

each sentence. In the Boundary Detection (BD) phase, F1 quantifies the system ability in recognizing the boundaries of each argument: every token (i.e. span) of every argument must be properly detected. In the Argument Classification (AC) phase, F1 measures the correctness of the role label assignment to each span. Table 7 reports the performance drop due to the error propagation in Babel. For example, if we consider the BD phase, a F1 score of 0.684 achieved in the FN setting with gold-standard information from the FP; it then drops to 0.560 when error propagates. Here, the most significant performance drops are observed in the BD and AC phase of the FN setting. It is a true bottleneck for system performance. This problem is overcome when HRI examples are used to re-training, as in the H setting that exhibit a stable improvement in F1. The benefits of the Hybrid setting can be seen in the sentence *grab the cigarettes next to the phone*. While the FN setting incorrectly classifies the arguments of the TAKING event, in `TAKING`(`theme`:[the,phone], `source`:nil), the H setting recognizes the more complete `TAKING`(`theme`:[the,cigarettes], `source`:[next,to,the,phone]).

Table 7. Babel chain analysis

Gold information at each step						
	FP		BD		AC	
	FN	H	FN	H	FN	H
GG	0.826	0.833	0.684	0.871	0.589	0.822
S4R	0.812	0.817	0.743	0.872	0.736	0.912
RC	0.732	0.758	0.696	0.817	0.701	0.898

Non-Gold information at each step						
	FP		BD		AC	
	FN	H	FN	H	FN	H
GG	0.826	0.833	0.560	0.680	0.373	0.612
S4R	0.812	0.817	0.598	0.712	0.502	0.688
RC	0.732	0.758	0.527	0.635	0.451	0.597

5 Conclusion

This paper proposes the study and comparison of two paradigms for the development of automatic Natural Language Understanding systems for Human Robot Interaction. First, we discussed the application of a grammar-based system. Then, a data driven approach, based on a free-from speech recognizer and a statistical semantic parser, is examined. We analyzed both solutions also from an engineering perspective, considering the system robustness as well as the effort of its realization and adaptability across different domains. As expected, experimental results show a good performance of the grammar-based method in controlled data. However a significant performance drop is experimented as the linguistic phenomena change and the complexity grows. This lack of robustness is compensated in the data driven approach. More important, this method enables a portable solution that significantly improves performance by adding a restrained number of annotated examples.

This is our first investigation in the Natural Language HRI field, but results clearly show possible research directions. In order to improve the overall system robustness, a deeper analysis of the speech recognition step is required as suggested by the results achieved in [3]. Moreover, the availability of more domain-dependent resources, as the corpora used in these experiments and presented in [5], resulted to be beneficial. The obtained results confirmed the importance of exploiting such datasets, allowing future HRI system to exploit them as a benchmark for a comparable evaluation. Other linguistic semantics theories should be investigated, e.g. Spatial Semantics especially to study the relationships between the output of the processing chain and the final command. Moreover, the role of visual features is here not considered, but it is certainly a future direction to be investigated.

REFERENCES

[1] Y. Altun, I. Tsochantaridis, and T. Hofmann, ‘Hidden Markov support vector machines’, in *Proceedings of the ICML*, (2003).

[2] Collin F. Baker, Charles J. Fillmore, and John B. Lowe, ‘The berkeley framenet project’, in *Proceedings of ACL and COLING*, Association for Computational Linguistics, (1998).

[3] R. Basili, E. Bastianelli, G. Castellucci, D. Nardi, and V. Perera, ‘Kernel-based discriminative re-ranking for spoken command understanding in hri’, in *AI*IA*, volume 8249 of *LNCS*, (2013).

[4] E. Bastianelli, D. Bloisi, R. Capobianco, F. Cossu, G. Gemignani, L. Iocchi, and D. Nardi, ‘On-line semantic mapping’, *Proceedings of ICAR*, (2013).

[5] E. Bastianelli, G. Castellucci, D. Croce, R. Basili, D. Nardi, and L. Iocchi, ‘Huric: the human robot interaction corpus’, in *9th edition of the LREC*, Reykjavik, Iceland, (2014). to Appear.

[6] Johan Bos, ‘Compilation of unification grammars with compositional semantics to speech recognition packages.’, in *COLING*, (2002).

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

[8] L. Carlucci Aiello, E. Bastianelli, L. Iocchi, D. Nardi, V. Perera, and G. Randelli, ‘Knowledgeable talking robots’, in *AGI*, volume 7999 of *LNCS*, (2013).

[9] 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*, pp. 859–865, (2011).

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

[11] James Curran, Stephen Clark, and Johan Bos, ‘Linguistically motivated large-scale nlp with c&c and boxer’, in *Proceedings of ACL*, Czech Republic, (June 2007). Association for Computational Linguistics.

[12] Juan Fasola and Maja J. Matarić, ‘Using semantic fields to model dynamic spatial relations in a robot architecture for natural language instruction of service robots’, in *IEEE/RSJ IROS*, Tokyo, Japan, (2013).

[13] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, Aditya A. Kalyanpur, A. Lally, J. William Murdock, E. Nyberg, J. Prager, N. Schlaefer, and C. Welty, ‘Building Watson: An Overview of the DeepQA Project’, *AI Magazine*, **31**(3), (2010).

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

[15] Thorsten Joachims, Thomas Finley, and Chun-Nam Yu, ‘Cutting-plane training of structural SVMs’, *Machine Learning*, **77**(1), 27–59, (2009).

[16] Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas Roy, ‘Toward understanding natural language directions’, in *Proceedings of the 5th ACM/IEEE*, HRI ’10, pp. 259–266, Piscataway, NJ, USA, (2010).

[17] Geert-Jan M. Kruijff, *A Categorial-Modal Logical Architecture of Informativity: Dependency Grammar Logic & Information Structure*, Ph.D. dissertation, Faculty of Mathematics and Physics, Charles University, Czech Republic, April 2001.

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

[19] C.D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*, MIT Press, 1999.

[20] Maja Popović and Hermann Ney, ‘Word error rates: Decomposition over pos classes and applications for error analysis’, in *Proceedings of StatMT ’07*, pp. 48–55, Stroudsburg, PA, USA, (2007). Association for Computational Linguistics.

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

[22] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer, ‘Feature-rich part-of-speech tagging with a cyclic dependency network’, in *In Proceedings of HLT-NAACL*, (2003).

[23] Adam Vogel and Dan Jurafsky, ‘Learning to follow navigational directions’, in *Proceedings of ACL ’10*, pp. 806–814, Stroudsburg, PA, USA, (2010). Association for Computational Linguistics.

[24] T. Wisspeintner, T. van der Zant, L. Iocchi, and S. Schiffer, ‘RoboCup@Home: Scientific competition and benchmarking for domestic service robots’, *Interaction Studies*, **10**(3), 393–428, (2009).