# Robust Spoken Instruction Understanding for HRI

Rehj Cantrell and Matthias Scheutz and Paul Schermerhorn and Xuan Wu
Human-Robot Interaction Laboratory
Indiana University
Bloomington, IN 47406, USA
Email: {rcantrel,mscheutz,pscherme,xuanwu}@indiana.edu

***Abstract*—Natural human-robot interaction requires different and more robust models of language understanding (NLU) than non-embodied NLU systems. In particular, architectures are required that (1) process language incrementally in order to be able to provide early backchannel feedback to human speakers; (2) use pragmatic contexts throughout the understanding process to infer missing information; and (3) handle the underspecified, fragmentary, or otherwise ungrammatical utterances that are common in spontaneous speech. In this paper, we describe our attempts at developing an integrated natural language understanding architecture for HRI, and demonstrate its novel capabilities using challenging data collected in human-human interaction experiments.**

***Keywords*-natural human-robot interaction; natural language processing; dialogue interactions; integrated architecture**

## I. Introduction

Most approaches to natural language understanding on robots (e.g., [1]–[4]) are sequential, make limited use of context, task knowledge, and goal structures, and ignore physical aspects of language users. In contrast, converging evidence from psycholinguistics suggests that human language understanding is incremental and parallel, depends on the speaker's and listener's contexts, utilizes task and goal knowledge, and involves the perceptions and perspectives of situated, embodied agents. While this difference in processing style may not matter for many NLU applications, it is critical in situations where humans and robots interact *naturally* as embodied agents co-located in the same environment [5].

In this paper, we describe our architecture for robust natural language understanding, which is a first attempt at meeting the functional requirements stated in the abstract. We start by briefly motivating our research and reviewing related work. Next, we describe an HRI task for which we have previously collected spoken dialogue data from human-human interactions, and describe several challenges for spoken language understanding on robots. We then provide an overview of the proposed robust NLU architecture for HRI, and demonstrate how it can handle various challenges of natural dialogue interactions. We give examples showing in detail how the system works, using the experimental data. Finally, we discuss current limitations and provide an outlook for future work.

## II. Motivation and Related Work

Spoken natural language understanding on robots in HRI contexts has two critical components that distinguish it from other NLU applications, due to the way humans interact in natural language: Component 1 (C1) human speakers expect interlocutors to produce backchannel feedback (e.g., in the form of eye gaze, verbal acknowledgments such as "okay" or "mhm", or through actions like head nodding) in order to communicate understanding during the utterance (e.g., [6], [7]); and Component 2 (C2) when listeners and speakers are co-located in the same environment, speakers expect them to rapidly and incrementally integrate perceptual context (e.g., in the resolution of reference, [8], [9]). While there is no current robotic system that even comes close to *natural* HRI, several recent efforts have tackled different aspects of these challenges and advanced the state of the art in natural language interactions of robots with humans.

C1 requires partial semantic interpretations and action execution based on partial meanings, and thus incremental NLU that respects human timing. Several recent NLU systems have addressed incremental processing at both syntactic and semantic levels [10]–[12], possibly even including pragmatic constraints [13], even though most of them have not been integrated into robotic architectures. And there are recent attempts to limit processing time to allow for fast natural language understanding (e.g., [14] propose a perceptron-based learning approach that allows an incremental parser to prune the set of possible parse trees after each word, addressing the real-time aspects of parsing on robots). Similarly, [15] tightly integrates natural language processing and action execution, which, in conjunction with incremental processing [16], is a prerequisite for backchannel feedback.

C2 requires incremental multi-modal information integration during utterance processing to constrain possible interpretations and intended meanings, again while respecting human timing – [17], for example, introduce a semantics-based approach on a robot to overcome errors by the speech recognizer, addressing the robustness aspect (both for disfluencies and speech recognition errors), and [18] integrate gaze information into an NLU system in order to disambiguate deictic expressions that are accompanied by a pointing gesture of the human, addressing some of the perceptual aspects.

While the above efforts have made significant advances in situated natural language understanding, there is currently no integrated HRI architecture that combines multiple of the above aspects together with handling common disfluencies. Yet, any robotic system that intends to allow for *natural* HRI in spoken language will have to meet the above human expectations, while simultaneously handling disfluencies that naturally occur in spontaneous speech.

## III. Challenges for Spoken Language Understanding on Robots

While no HRI architecture is currently capable of handling completely unrestricted natural language, we will show that it is reasonable and possible to handle fairly unconstrained natural language produced by humans by constraining the *task*. The advantage of a constrained task is that both the number of concepts and skills required to perform it, as well as the number of words and linguistic expressions needed to coordinate activities with other humans, are very limited.

We conducted experiments in which subjects performed a *team search task* [19] in order to collect data about the types of natural language exchanges that humans are likely to produce in a task of relevance to HRI. In this task, two humans, the *director* and the *member*, located in different physical spaces, were required to coordinate their actions using natural language via a wireless audio link in order to accomplish several goals within a limited amount of time: (1) the director had a partial map of the search environment and was to direct the member (located in the search environment) through the environment to the location of a cardboard box, which the member was to retrieve; (2) the director was to assist the member with finding 8 blue boxes located in the search environment by giving directions based on the information on the map; (3) the member was then to empty colored blocks contained in the blue boxes into the cardboard box, leaving the blue boxes in their location; (4) in parallel, the member was to tell the director the location of each of 8 green boxes that were also positioned within the search environment (but not indicated on the director's map); (5) five minutes into the task, a new goal was communicated to the director that required the member to place one yellow block from the cardboard box into each of the 8 pink boxes (also located in the environment). The task ended after 8 minutes.

12 dialogues from the human interactions in the search task have been transcribed, totalling 2,683 utterances (with 1,637 if repetitions are removed), comprising 17,783 tokens and 712 types. Analysis of the data clearly demonstrates the need in HRI for a robust NLP system that can handle the following:

- ungrammatical sentences (incomplete referential phrases, missing verbs, corrections, and others)
- wrong word substitution for intended target words (e.g., "block" and "book" for "box")
- underspecified directions, referents, and directives (which require shared task-knowledge and knowledge of sub-goals)
- "ums" and "uhs" (that indicate cognitive load)
- coordinating "okays" and other forms of verbal backchannel feedback (that indicate dialogue moves)

## IV. Functional Requirements for Robust Language Understanding in the Search Task

In this section, we describe some of the important functional requirements concerning *disfluencies*, *navigational directions*, *omissions*, *anaphora resolution*, and *back-channel feedback* that a robotic NLU system will have to deal with, even in a limited HRI domain such as our search task. All examples are taken from the human-human dialogue corpus described above.

### A. Disfluencies

First and most obvious, the system must recover from disfluencies, including:

- non-lexical fillers (e.g. *um* `let me get that green box`)
- lexical fillers (e.g. `I 'm just` *like* `next to the door frame`)
- abandoned utterances (e.g. *next to the in h–*)
- and repairs or corrections (e.g. `so how many b–` `how many` *box– blue boxes* `do we have?`)

These disfluencies cause several problems. First, they severely impact speech recognition, as speech recognizers must recognize vocabulary items, but the number of possible disfluent words is much too great to include in such a vocabulary. Without those words in the vocabulary, however, the recognizer will attempt to recognize these out-of-vocabulary words as vocabulary words (not only may it recognize them as separate words themselves, they are often recognized as being part of a larger incorrect word, and thus their influence spreads even to non-disfluent items).

Second, they can prevent correct semantic interpretation, particularly in cases of abandoned utterances and repairs/corrections. E.g., the utterance `I'm just like next to the door frame` could be (wrongly) interpreted as containing the preposition `like`, indicating that the speaker is in a situation similar to being next to the door frame.

### B. Navigational directions

While it is fairly straightforward to keep track of named locations such as "the room" or "the hallway", the human-human data indicates that participants tend to use descriptions based on observed landmarks: `so go back into the room um right– go r– go where you're right between the cubicle and the filing cabinet`. Correctly processing an utterance like this requires significant perceptual abilities (and conceptual knowledge), as well as tight integration of the NLU and perceptual systems.

### C. Omissions

Humans often omit or condense information. For example, in `well get the yellow block from there and then in that room` the robot needs to understand that there are two yellow blocks to get: one from `there` and one from `that room`.

### D. Grounding of referents

In the previous example, the robot needs to understand the referent of `that room`. There are two parts to this problem: *anaphora resolution* and *real-world grounding*. In this context, *anaphora resolution* refers to the connection of different verbal

references to the same abstract entity, while *grounding* refers to the connection of that abstract entity to an entity in the world. In this paper, we focus on the second part, to allow the robot to carry out activities on real-world entities using abstract information from the verbal domain.

In the instruction `so now you have to get the blue things right`, the term `things` must be resolved as referring to boxes because the only blue items were boxes, and the speaker had been told they were to collect blue boxes. However, blue boxes are not previously mentioned in the dialogue, thus ordinary anaphora resolution techniques of attempting to connect this phrase to previous references to the same entity will fail. Instead, *task-based* information must be used in order to correctly resolve the reference.

### E. Back-channel feedback

For robust natural language processing, it is essential that a system be able to continue listening and processing while it is speaking. For example, in the following dialogue:

```
M: two yellow blocks?
D: yeah there should be-
M: can they be from different boxes?
D: well there should just be one yellow block in each blue box
```

the member interrupted the director in order to ask a question, and the director stopped in mid-utterance to answer the question. If the director had continued speaking, ignoring the question, the interaction would not have been effective. Similarly, the robot should be able to speak while it is listening. Humans often provide verbal feedback in order to show that they are understanding what is being said:

```
D: so turn right
M: kay
D: and walk a little bit and turn right again
M: kay
D: walk down that
```

The member responded after each segment of the instructions as the director was providing them. When this feedback is not provided, humans become unsure of whether they are communicating effectively. In the dialogue sequence:

```
M: and then the second green box the number two
D: yeah
M: um is on the second step of the wooden platform
```

the director provides feedback after the noun phrase as the member describes the location of a box. In some cases, the director's response is complete before the member's query finishes:

```
M: do we back track to the pink in the hallway where I was?
D: yeah
```

While it is probably not necessary for a robot to be capable of replicating the most extreme examples of backchannel feedback, it should be able to provide some cues to indicate that it is following the conversation.

## V. An Integrated Architecture for Robust Spoken Instruction Understanding

Our proposed architecture integrates *speech recognition*, *incremental parsing*, *incremental semantic analysis*, *disfluency analysis*, and *situated reference resolution* components into our robotic `DIARC` architecture [5], which provides mechanisms for integrating incremental natural language processing components with action execution [16], [20]. The present work expands the capabilities of the previous work with the inclusion of a more sophisticated parsing mechanism, allowing greater flexibility and more robust natural language understanding.

### A. Speech recognition

In current speech recognition systems, there is a trade-off to be made: one must either train on individual users to be able to recognize a wider range of utterances, or one may keep speaker-independence at the cost of being able to recognize a much smaller range of utterances. We have chosen the latter approach, using the CMUSphinx speech recognition system,[1] in conjunction with a CMU acoustic model (WSJ) and a hand-crafted finite-state grammar to reduce word error rates.

It should be noted that the use of a finite-state grammar in the speech recognizer is limiting, especially when it comes to handling natural disfluencies. In order for the speech recognizer to pass on those disfluencies, they must be included in the grammar. But it is difficult to enumerate all possible (or even probable) disfluencies, which limits the extent to which we can evaluate the handling of natural disfluencies in subsequent processing stages (e.g., syntactic and semantic approaches). In the future, we will explore such options as using a speaker-dependent model and an n-gram language model, which would allow for a much larger array of possible utterances, and tuning our disfluency handling for speech recognition data, including recognition errors.

### B. Incremental parsing

We selected an incremental, data-driven dependency parsing algorithm because it is fast and works by creating links (*dependencies*) between individual words (the *head* and the *dependent*), rather than forming and linking phrases. Operating at the level of words makes it ideal for translating each head-dependent relationship into a lambda-logical semantic expression taking an argument (as described below). The algorithm was adapted from MaltParser, a shift-reduce dependency parser [21] that maximizes the probability over individual actions (rather than over a full utterance). *Incrementality* refers to the notion that the parser should always have a complete working parse of whatever has been input. So, while most parsers (including MaltParser) produce no output until the entire sentence has been analyzed, an incremental parser has a parse even in the middle of a sentence. Using this intermediate parse, we can begin semantic analysis even before the entire sentence has been uttered.

[1] http://cmusphinx.sourceforge.net/sphinx4/

Nivre showed that the dependency parsing algorithm was ideal for incremental parsing [22]. Our implementation, Mink, is the first actual incremental implementation we know of. The algorithm itself is much like its constituent-based forerunner. The shift and reduce actions are the same, and two new actions, left-arc and right-arc, are introduced. Left-arc creates a dependency arc pointing from the head on the input to the dependent on the stack, whereas right-arc creates an arc pointing from the head on the stack to the dependent on the input. The Logistic machine learning algorithm from the Weka library[2] is used to decide the most probable next action.

We trained our parser on 90 percent of the experimental corpus, and tested on the 10 remaining percent. The data had been filtered for disfluencies. Our accuracy was 90%. We also tested the full system with the incremental parser vs. the full system with a non-incremental parser using the same algorithm, and showed that the incremental version took, on average, 35% of the time taken by the nonincremental version. A preliminary description and evaluation of Mink is presented in [23].

### C. Incremental semantic analysis

The semantic analysis performed during syntactic parsing uses lambda conversions to attach logical forms to lexical items, forming semantic interpretations of sentence fragments. In order to facilitate the conversion, we used the well-established connection between lambda representations and combinatorial categorial grammar (CCG) [24]. As each dependency arc was added, we constructed CCG tags consisting of each predicate's return type and arguments, and then used the combinatorial properties of the CCG grammar to perform lambda reduction. Note that CCG tags give significantly more information than simple part-of-speech tags, namely *return type and arguments*. The return type is basically the part-of-speech tag, while the arguments are similar to phrasal constituents. Our method of constructing CCG tags from dependency arcs, therefore, involves two steps: determining the return type and finding the arguments.

When a new token input arrives, an "empty" CCG tag is created, including the return type and a (possibly empty) list of arguments. Currently, the return type is determined immediately upon creation of the new tag in order to reduce the complexity of the computation. This is posed as a classification problem. The features used to make the classification are currently the token and the WSJ POS tag returned by a tagger. Whenever a dependency arc is discovered with a given token as the head, the dependent's return type is added as an argument to the head's CCG tag.

For example, the phrase `the blue box` might be tagged and parsed as follows by a constituent parser using the WSJ tagset:

$$(NP(the_{DT}\ blue_{JJ}\ box_{NN})).$$

However, with a CCG parser, it would be tagged and parsed as

$$(NP\ (the_{NP/NP}\ (NP\ blue_{NP/NP}\ (NP\ box_{NP}))).$$

The CCG representation of the parse graph is then used in the semantic conversion to logical form. This conversion uses a semantic dictionary to translate from parse trees to semantic representations: each lexical entry in the dictionary consists of a word, its possible CCG tags, and all of its lambda-logical expressions for each word/CCG tag combination. The goal of the conversion is to do the following:

- translate instructions into appropriate actions that can be carried out by the robot
- generate semantically meaningful units that can be incrementally grounded
- in conjunction with (2), identify points where backchannel feedback in the form of verbal acknowledgments or eye movements is appropriate

Our previously developed semantic analysis component could, based on the order dictated by the CCG grammar, combine *temporal and dynamic logic* formulas with $\lambda$-expressions to create $\lambda$-free temporal and dynamic logic expressions representing the goals and actions specified by the natural language instructions [25]. The present work builds on that two-pronged approach of extracting both goals and actions at the same time. Performing semantic processing *incrementally* provides many advantages (beyond what we are able to discuss here): (1) it allows the robot to respond quickly when it does not understand an expression (cp. [16]); (2) it enables the robot to check the consistency of the new goals from the directives with existing goals; (3) it gives the robot an (at least partial) action sequence to achieve the goals (which can be further refined via standard task planners if necessary); and (4) it allows the robot to detect and react quickly to important syntactic, semantic, and pragmatic ambiguities in the instruction (e.g., it could initiate a head movement if referents mentioned in the utterance are not currently visible). We will give examples of the conversion process in the Demonstration section.

### D. Disfluency analysis

Disfluencies are targeted for correction or repair by a specific method based on their types.

- *Non-lexical disfluencies* are removed by a regular expression-based filter.
- *Lexical disfluencies* require that the system be able to successfully differentiate between meaningful and disfluent uses of words (e.g., "like"). Our system uses a trigram statistical model that labels such words as disfluent or not based on a language model.
- *Abandoned utterances* (e.g., when a speaker begins a particular utterance, then abandons it to switch to a different conversational segment) are challenging to detect and recover from, particularly when using a robust statistical parser. In our system, these tend to fall out simply by not being fully *semantically* parsable.
- *Repetitions* are handled similarly to non-lexical disfluencies, using simple rule-based methods to remove repeated strings of words up to a maximum length of 5 words.

[2]http://www.cs.waikato.ac.nz/ml/weka/

- *Repairs and corrections* in utterances are sometimes made distinct by repair markers such as "I mean" or "wait", but that is often not the case. A set of repair markers was extracted from the human-human experiment corpus; when a word or phrase of one type is separated from another word or phrase of the same type by one of these markers, the latter is taken to be a replacement for the former. Repairs that are not signaled by a marker typically lead to failed semantic parsing, as in the case of abandoned utterances.

These methods chained together constitute the disfluency analysis subsystem. First, non-lexical disfluencies are removed by a regular expression filter. Next, lexical disfluencies are identified by the trigram language model and removed (Table I).

TABLE I
REMOVING LEXICAL AND NON-LEXICAL DISFLUENCIES

| Input | Output |
|---|---|
| *um* let me get that green box hold on | let me get that green box hold on |
| *so now* you have to get the blue things *right* | you have to get the blue things |
| i 'm just *like* next to the door frame | i 'm just next to the door frame |
| *like* right in front of it | right in front of it |
| *well* get the yellow block from there and then in that room | get the yellow block from there and then in that room |

When repair markers are identified, the system checks for matching phrase types on either side to treat as replacements. In each of the phrases

- `there 's` **`a blue kit`** `I mean` **`a blue box`** `at the end of the table`
- `go back to` **`the hallway`** `er` **`that little room`**
- `go into that first small room that would be on` **`your righthand`** `er` **`your lefthand side`**

the repair marker signals the disfluency module to replace the first bold-face phrase with the second. While this method is effective in many cases, it does have limitations. For example, "yes" is generally used as a positive emphasizer, as in

`a stack of boxes yes get that one`
`cart yes there is a cart.`

However, in `so` **`there are no blue boxes`** `oh yes there is` **`there 's a blue box behind the door`** it is part of a repair marker, but is unique in the corpus and hence not included in the list of repair markers. Replacement would, therefore, not be triggered, and the system would have no way of resolving the conflict without asking the speaker.

Finally, the parser attempts to identify CCG tags that have been generated with extra and doubled arguments. For example:

`how many b how many box blue boxes do we have`

is parsed (in part – irrelevant details are omitted for simplicity) as:

$[_{OBJ}[_{WRB}how\ many[_{NP}b]$
$[_{OBJ}[_{WRB}how\ many[_{NP}box][_{NP}blue\ boxes]]do]]we\ have$

The first omission can be made in the place where `how many` has been connected to two noun phrases. The second one is selected on the assumption that it is a correction:

$[_{OBJ}[_{WRB}how\ many[_{NP}b]$
$[_{OBJ}[_{WRB}how\ many[_{NP}blue\ boxes]]do]]we\ have$

Next, two objects have been selected for the verb `do`. Again the first one is discarded:

$[_{OBJ}[_{WRB}how\ many[_{NP}blue\ boxes]]do]]we\ have$

The final result,

`how many blue boxes do we have`

is now in a syntactic form that can be processed semantically. Similarly, our previous example

`so go back into the room um right- go r- go where you're right between the cubicle and the filing cabinet`

becomes the slightly more tractable

`go back into the room go where you're between the cubicle and the filing cabinet.`

A limitation of the system is that it cannot handle *aborted words*, which are a very common disfluency type and by far the most difficult one to detect. While NLU systems that rely on human transcriptions for disfluency detection and repair can simply remove aborted words prior to automatic processing, this approach is clearly ill-suited in situated, embodied NLU contexts where robots must handle actual speech recognition output. There are two likely outcomes when a typical speech recognizer encounters an aborted word: it will either be misrecognized as a short in-vocabulary word, or, worse yet, it may be misinterpreted as part of a longer word. Because our current speech recognition configuration employs a finite state grammar (which would require each possible aborted word to be included in the grammar), aborted-word disfluencies are not passed to the NLU system. Sophisticated acoustic handling methods could be used to detect aborted words, but such methods are outside the scope of this paper.

### *E. Situated reference resolution*

There are two steps to situated referent resolution: (1) anaphora resolution (finding out what noun is referred to by a pronoun) and (2) grounding (finding out what real-world referents the nouns refer to). We use a combination of recently-attended-to entities in conjunction with a search tree in order to determine whether task-based information is needed for grounding. Specifically, we have a table of "last" objects (e.g., "last mentioned object", "last visual object") in addition to task-based information (e.g., we're looking for blue boxes), followed by a method of searching for each type of object (e.g., visual search for blue boxes). A decision tree directs the search

when grounding is performed. For example, to ground the phrase `blue things`, the table of "last objects" is searched to see whether any blue object has been mentioned. If the word `blue` has not been previously used in the present discourse, this search will fail. The task information branch of the tree, in which all entities defined in the task description are listed, is then searched. In this example, the task of finding '`blue boxes` would match. In cases where the referent has not been previously mentioned and is not part of the task knowledge, the process falls back on a sensory search based on the modality associated with the referent's type. In this case, because the content word of the phrase involves a color, the grounding process would have triggered a visual search for blue objects.

## VI. Demonstration

Section V described the system and provided examples of how individual components function. However, since we are not concerned with component functionality or performance in isolation (e.g., accuracy numbers of parses), we need to demonstrate the integrated system on a robot in a relevant real-world task. For this purpose, we again selected examples from the human-human corpus that demonstrate two important challenges for spoken instruction understanding on robots in HRI scenarios: (1) semantic ambiguities and (2) incremental understanding with backchannel feedback. The demonstration platform was a Pioneer P3AT equipped with a Bumblebee Firewire camera for vision processing and a dual-core Linux PC for onboard processing. The experimental environment was a hallway similar to the environment in which the human-human experiments were conducted (see Figure 1). Some of the instructions required actuating capabilities that were not available on the physical robot (e.g., to collect boxes); these examples were demonstrated using a 2D real-time physical simulation of the physical environment that the architecture can run on identically to the physical robot without any changes (in fact, the architecture has no way of knowing whether it controls the physical or the simulated robot).

### A. Example 1: Semantic Ambiguity

We use the instruction `so now you have to get the blue things right` from our corpus and consider two variants:

(1a) `you have to get the blue thing`
(1b) `you have to get the blue things`

Note that the only overt syntactic difference between (1a) and (1b) is the number of the final word: `thing` is singular in the former and plural in the latter. However, the semantic differences are much greater. The first sentence is a command to get a specific object. If a human looking at a table with several objects, two of which are blue, and were instructed to get `the blue thing`, then the human – in the absence of other identifying information – would most likely request clarification (similarly, the human would request clarification if no blue object were present). Hence, it is clear that `the blue thing` must refer to exactly one object. However, `the blue things` refers instead to a *set* of objects, which must be interpreted as having been either pre-defined, or being limited by perceptual or other contextual (e.g., task-based) constraints.

Fig. 1. A snapshot from the demonstration setup on the robot in the hallway (top) and in the 2D physical simulation (bottom).

TABLE II
Singular and Plural Lambda Expressions

| word | number | definition |
|---|---|---|
| the | sing. | $\lambda Y \exists x.Y(x) \wedge \forall y Y(y) \rightarrow x = y$ |
| the | pl. | $\lambda Y\{x \vert Y(x)\}$ |
| blue | | $\lambda Z \lambda z.blue(z) \wedge Z(z)$ |
| thing | sing. | thing |
| things | pl. | thing |

This distinction makes the lambda expressions capturing the two meanings quite different (here we focus on the referential aspect expressed in a fragment of first-order logic augmented by simple set constructs):

(a) $\exists x(blue(x) \wedge thing(x) \wedge \forall y(blue(y) \wedge thing(y) \rightarrow x = y))$
(b) $\{x \vert blue(x) \wedge thing(x)\}$

(a) means that there is exactly one blue thing in the discourse context, whereas (b) refers to the set of all blue things. The

TABLE III
LAMBDA EXPRESSIONS AND ASSOCIATED CCG TAGS

| word | number | definition |
|---|---|---|
| the | NP/N | $\lambda Y \exists x.Y(x) \wedge \forall y(Y(y) \rightarrow x = y)$ |
| the | NP/NN | $\lambda Y\{x\|Y(x)\}$ |
| blue | N/N | $\lambda Z \lambda z.blue(z) \wedge Z(z)$ |
| blue | NN/NN | $\lambda Z \lambda z.blue(z) \wedge Z(z)$ |
| thing | N | thing |
| things | NN | thing |

distinction poses challenges for incremental parsing, as the difference must be generated already at `the`, which is not explicitly marked for number (see Table II for the lambda terms attached to the lexical items and Table III for the same with CCG tags). In such cases, both interpretations are pursued until one fails (cp. to [16]). Note that in order to have the noun's number feature carried through the adjective to the determiner, the adjective's definition must also have a number feature, even though the definition of the adjective itself does not differ whether taking a singular or plural argument. Hence, when `the` is input, the parser retrieves both definitions, having no evidence as to which to select. Next `blue` is input, and again the parser retrieves both definitions and performs a lambda conversion. The two possible semantic interpretations at this point are:

(a) (NP/N) $\lambda Z \exists x.blue(x) \wedge Z(x) \wedge \forall y(blue(y) \wedge Z(y) \rightarrow x = y)$.

(b) (NP/NN) $\lambda Z\{x|blue(x) \wedge Z(x)\}$

Finally, either `thing` (N) or `things` (NN) is received, which, based on the CCG tags, decides which reduction is retained. The reductions in turn generate actions. The singular version generates the action "get" with the argument "blue" (causing the robot to get the single blue box based on finding the single referent that makes the definite description true), whereas the plural version generates the action "getAll" (which causes the robot to search for and retrieve all blue boxes in its environment). In our tests, the robot was able to successfully understand both spoken instructions and perform the appropriate actions corresponding to the singular and plural variants.

### *B. Example 2: Incremental Understanding with Backchannel Feedback*

The system's ability to generate appropriate backchannel feedback was demonstrated with the example `turn right and walk a little bit and turn right` (Table IV shows the associated lambda terms). In general, listeners tend to provide verbal feedback only between phrases, to demonstrate understanding and remain conversationally aligned with the speaker. As described in Section IV-E, the member provided feedback after the first `turn right` and then responded with an acknowledgment after the utterance. However, the member could just as easily have provided feedback after `walk a little bit`. For the purposes of the present demonstration, both are marked as possible feedback points. We leave aside, for the time being, the question of how the robot should choose at which, of all possible feedback points, to actually provide the feedback. We also leave aside the question of how to determine the best, or most appropriate, form of feedback (e.g., whether to nod, say "uh huh", or provide some other response).

TABLE IV
LAMBDA CONVERSIONS

| Word | CCG | Definition |
|---|---|---|
| turn | S/DIR | $\lambda x.facing(x)$ |
| right | DIR | right |
| and | $S\backslash S/S$ | $\lambda x \lambda y(x; y)$ |
| walk | S/DIST | $\lambda x.walk(x)$ |
| a-little-bit | DIST | small_random_distance |

Note that `a-little-bit` is taken to be a primitive. This is because the architecture has no mechanism for deciding what is appropriate when faced with relative and ambiguous expressions such as `a little bit`, which could refer (as it does here) to a distance, to a concrete amount ("I'd like a little bit of coffee, please"), to an abstract amount ("I just have a little bit left to do on my paper"), or to any number of other types of things. Each of these types of meanings should receive different tags so they can be differentiated.

Another word that can have several meanings is "and": two of those meanings are *parallel "and"* and *sequential "and"*. In the former, the robot would perform actions in parallel. For example `go to the end of the hallway and get the blue boxes` should generate the actions in parallel so that the robot would gather blue boxes on its way to the end of the hallway. On the other hand, `turn right and go forward` should most likely generate the sequence of events, `first turn right, then go forward`. We use that sentence to demonstrate how the system handles instances of the *sequential "and"*.

The first connection we find is between `turn` and `right`. The combination generates the action term `turn(right)`. At this point, the system has a completed combination and is immediately able to understand what to do. When the word `and` is received, a connection is made between it input and the previous output, yielding $\lambda y.turn(right); y$. (where ";" denotes the action composition in our dynamic logic, see [26]). At this point the system can determine that it must wait for further instructions. The remaining inputs (`walk` and `a-little-bit`) are combined to create another action term, and the connection between it and the `and` allows the system to reduce to form the combination $turn(right); move(small\_random\_distance)$. Again we have a completed action specification and the robot could provide feedback. Additional (analogous) combinations lead to the final output $turn(right); move(small\_random\_distance); turn(right)$, and the final response point is marked. At any of three response points, the robot can choose to verbally indicate understanding; in the experimental runs, we modeled the human data by providing feedback only after the first segment.

Practically, the current architecture does not have any way of knowing what the real endpoint of the utterance is. Theoretically, the robot could mark the last point and then continue waiting for further instructions indefinitely (and of course, the speaker could continue to give instructions indefinitely, while the robot patiently interpreted them). In the future, intonation could be used to help determine whether the speaker is finished or not.

## VII. Discussion and Conclusion

In this paper, we presented our integrated robotic architecture for robust natural language understanding and demonstrated its functionality and performance based on dialogue data collected previously in human-human experiments in a relevant HRI search task. Overall, the proposed mechanisms are now able to handle a variety of challenging spoken utterances in real-time on a robot, including various forms of disfluencies and ambiguities that require task-based reference resolution as exhibited by humans. Thus, we view the current architecture as an important step in the direction of achieving natural human-robot interactions.

Given the encouraging results from the corpus-based demonstrations presented here, we believe that it is feasible to test the architecture in the search task with human subjects in the near future, with the robot assuming the role of the "member" being guided by instructions from the human "director". These experiments will provide a quantitative evaluation of the architecture, while simultaneously allowing us to compare how humans speak and behave when they are interacting with humans versus robots.

In addition, we are interested in further reducing speech recognition errors by comparing speech recognition output with human transcripts and finding ways to compensate for the differences. And we are also interested in addressing the problem that all semantic expressions in the current system are hand-crafted, by integrating lambda learning mechanisms that use context in conjunction with known conceptual definitions to induce meanings for unknown words.

## Acknowledgment

This work was in part funded by ONR MURI grant #N00014-07-1-1049 to the second author.

## References

[1] M. P. Michalowski, S. Sabanovic, C. DiSalvo, D. B. Font, L. Hiatt, N. Melchoir, and R. Simmons, "Socially distributed perception: GRACE plays social tag at AAAI 2005," *Autonomous Robots*, vol. 22, no. 4, pp. 385–397, 2007.

[2] R. Müller, T. Rofer, A. Landkenau, A. Musto, K. Stein, and A. Eisenkolb, "Coarse qualitative description in robot navigation," in *Spatial Cognition II*, C. Freksa, W. Braner, C. Habel, and K. Wender, Eds. Berlin: Spinger-Verlag, 1998, pp. 265–276.

[3] R. Moratz, K. Fischer, and T. Tenbrink, "Cognitive modeling of spatial refernce for human-robot interaction," *International Journal on Artificial Intelligence Tools*, vol. 10, no. 4, pp. 589–611, 2001.

[4] K. Wauchope, "Eucalyptus: Integrating natural language input with a graphical user interface," Naval research Laboratory, Tech. Rep. NRL/FR/5510-94-9711, 1994.

[5] M. Scheutz, P. Schermerhorn, J. Kramer, and D. Anderson, "First steps toward natural human-like HRI," *Autonomous Robots*, vol. 22, no. 4, pp. 411–423, May 2007.

[6] K. Eberhard, M. Spivey-Knowlton, J. Sedivy, and M. Tanenhaus, "Eye movements as a window into real-time spoken language comprehension in natural contexts," *Journal of Psycholinguistic Research*, vol. 24, pp. 409–436, 1995.

[7] D. Schiffrin, *Discourse Markers*. Cambridge University Press, 1988.

[8] H. Clark and C. Marshall, "Definite reference and mutual knowledge," in *Elements of discourse understanding*, A. K. Joshi, B. L. Webber, and I. A. Sag, Eds. Cambridge: Cambridge University Press, 1981, pp. 10–63.

[9] S. Brennan, "Centering attention in discourse," *Language and Cognitive Processes*, vol. 19, no. 10, pp. 138–167, 1995.

[10] D. DeVault and M. Stone, "Domain inference in incremental interpretation," in *Proc. ICoS*, 2003.

[11] J. F. Allen, B. W. Miller, E. K. Ringger, and T. Sikorski, "A robust system for natural spoken dialogue," in *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, A. Joshi and M. Palmer, Eds. San Francisco: Morgan Kaufmann Publishers, 1996, pp. 62–70.

[12] S. Varges and M. Purver, "Robust language analysis and generation for spoken dialogue systems," in *Proceedings of the ECAI workshop on Development and Evaluation of Robust Spoken Dialogue Systems for Real Applications*, Riva del Garda, Italy, Aug. 2006.

[13] W. Schuler, S. Wu, and L. Schwartz, "A framework for fast incremental interpretation during speech decoding," *Computational Linguistics*, vol. 35, no. 3, pp. 313–343, 2009.

[14] P. Lison and G.-J. M. Kruijff, "Efficient parsing of spoken inputs for human-robot interaction," in *Proceedings of the 18th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN09)*, Toyama, Japan, 2009.

[15] R. J. Firby, "Adaptive execution in complex dynamic worlds," Ph.D. dissertation, Yale University, 1989.

[16] T. Brick and M. Scheutz, "Incremental natural language processing for HRI," in *Proceedings of the Second ACM IEEE International Conference on Human-Robot Interaction*, Washington D.C., March 2007, pp. 263–270.

[17] S. Hüwel and B. Wrede, "Spontaneous speech understanding for robust multi-modal human-robot communication," in *Proceedings of the COLING/ACL on Main conference poster sessions*. Association for Computational Linguistics, 2006, pp. 391–398.

[18] Z. Prasov and J. Chai, "What is in a gaze? The role of eye-gaze in reference resolution in multimodal conversational interfaces," in *ACM 12th International Conference on Intelligent User Interfaces*, 2008.

[19] M. Scheutz and K. Eberhard, "Towards a framework for integrated natural language processing architectures for social robots," in *Proceedings of the 5th International Workshop on Natural Language Processing and Cognitive Science*, Barcelona, Spain, June 2008, pp. 165–174.

[20] T. Brick, P. Schermerhorn, and M. Scheutz, "Speech and action: Integration of action and language for mobile robots," in *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, October/November 2007, pp. 1423–1428.

[21] J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryiğit, S. Kübler, S. Marinov, and E. Marsi, "MaltParser: A language-independent system for data-driven dependency parsing," *Natural Language Engineering*, vol. 13, no. 2, pp. 95–135, 2007.

[22] J. Nivre, "Incrementality in deterministic dependency parsing," *Incremental Parsing: Bringing Engineering and Cognition Together. Workshop at ACL-2004*, 2004.

[23] R. Cantrell, "Mink: An incremental data-driven dependency parser with integrated conversion to semantics," in *Proceedings of RANLP Student Research Workshop 2009*, 2009.

[24] M. Steedman, *The syntactic process*. MIT Press, 2000.

[25] J. Dzifcak, M. Scheutz, C. Baral, and P. Schermerhorn, "What to do and how to do it: Translating natural language directives into temporal and dynamic logic representation for goal management and action execution," in *Proceedings of the 2009 International Conference on Robotics and Automation*, Kobe, Japan, May 2009.

[26] R. Goldblatt, "Parallel action: Concurrent dynamic logic with independent modalities," *Studia Logica*, vol. 51, no. 3/4, pp. 551–578, 1992.