Recognizing Intention from Natural Language: Clarification Dialog and Construction Grammar

Sean Trott
International Computer
Science Institute
Berkeley, CA 94704
Email: seantrott@icsi.berkeley.edu

Manfred Eppe
International Computer
Science Institute
Berkeley, CA 94704
Email: eppe@icsi.berkeley.edu

Jerome Feldman
International Computer
Science Institute
Berkeley, CA 94704
Email: feldman@icsi.berkeley.edu

***Abstract*—Integral to recognizing human intentions is understanding language. We describe an implemented system for Natural Language Understanding (NLU) that receives text or speech as input and produces action as output without human intervention. Using previous research from linguistic pragmatics and dialog systems, we identify several areas of difficulty in recognizing intentions from language. We discuss our implemented solutions to two of these problems: 1) clarification dialog; 2) a construction grammar approach to the interpretation of indirect speech acts.**

***Index Terms*—natural language understanding, human-robot interaction, clarification dialog, indirect speech acts, intention recognition**

I. Introduction

Although humans are adept at communicating intentions to other humans, and also at recognizing such intentions, facilitating this aspect of Human-Robot Interaction (HRI) has proved very complex. One reason for this difficulty is that humans communicate their intentions through a variety of modalities, including: posture, gesture, intonation and inflection, eye gaze, and language. In this work, we narrow the problem to intentions that are communicated through language. We also focus primarily on the machine recognition of human intentions, though we do discuss bidirectional communication between humans and machines.

Two major problems of intention recognition from language are:

1) Under-specified or ambiguous input (e.g. ”move the box north” when there is more than one box).
2) Utterances in which the intended effect is distinct from a literal interpretation. As an example, consider indirect speech acts like ”Can you move the box north”. Here, it is more likely the user intends a robot to actually move a box, not answer whether or not it is able to push the box.

Both problems occur in ordinary human discourse, but become significantly more pronounced in interactions between humans and robots. We suggest that research from cognitive linguistics can help address these issues.

In Section II, we discuss previous work on intention detection in language. In Section III, we provide an overview of the architecture for our NLU system that is capable of performing intention detection. In Section IV, we further investigate the problems outlined above, as well as our implemented solutions. Finally, in Section V, we suggest future applications and directions for this research on intention recognition.

II. Previous Work

To put our wok in context with the state of the art, we present a survey on related work on intention detection. Since the most important application domains of natural language understanding are in the field of autonomous systems, robotics, and similar applications, we will focus on these fields. Towards this, we follow the study in [16], which summarizes the most recent prominent approaches for NLU in robotics [12], [31], [32], [47], [26], [25], [39], [38], [35], [7], [10], [41], [42], [15], [45], [13], [2], [3], [4], [27]. The authors suggest that surprisingly little work in this area exists, and that intention detection and understanding indirect speech acts are among the most difficult problems in natural language understanding. In the following, we summarize those approaches that are related specifically to dialog and intention detection, and we also consider additional intention detection related literature in this brief survey. For brevity, we focus only on those approaches that use natural language for intention detection.

The work on the *DIARC* HRI-System [31], [12] has been extended in [47] to cover intention detection and specifically indirect speech acts. The authors use a Dempster-Shafer theoretic approach “ ...for inferring intentions I from utterances U in contexts C, and, conversely, for generating utterances U from intentions I in contexts C.” The approach is computationally tractable, and it includes also the generation of clarification dialogues. Our work differs from that of [47] in that we focus on the linguistic side of intention detection and, at the current state of our work, not yet on contextual background information.

The authors of [26], [25] focus on dialog. They represent the semantics of an utterance in a categorical modal-logical form, based on Combinatory Categorical Grammar (CCG) [36]. Their system can perform reference resolution and starts a clarification dialog if the reference is too ambiguous. For disfluency analysis, the authors use contextual knowledge to prime utterances. Verbal feedback with words like “okay” or “fine”, can also be handled. The authors present an implementation which they use to perform experiments with impressive results, but do not demonstrate how the system is connected to real or simulated robots in a modular manner.

The authors of [41], [42], [15], [45] pursue an information-theoretic approach to minimize uncertainty in language through asking appropriate disambiguating questions. The focus is on dialog and clarification. This could probably also be extended to detect intentions from indirect speech acts. The system is based on learning and so-called Spatial Description Clauses and Generalized Grounding Graphs [41].

The authors of [13] tackle the problem that humans and robots do not have a common perceptual ground because object recognition capabilities of robots are far behind human level. The work investigates how much and which additional effort is required to establish a common perceptual ground between robot and human. This can be a very important approach also for dialog and intention detection, but the authors do not address this issue.

The work presented in [8] is another approach that focuses on intention detection from context, but in contrast to [47], the authors do not use external background knowledge. Instead, they use two other different methods for this task. Firstly, they use previous sentences in a longer dialogue or text to infer intention, and secondly, they focus on detecting significant features. Another key aspect of their work is that the authors explicitly focus on spoken language and the problems that arise with speech recognition. The work considers interesting and imortant issues, but we believe that using a richer language model like our ECG approach could further boost intention detection capabilities.

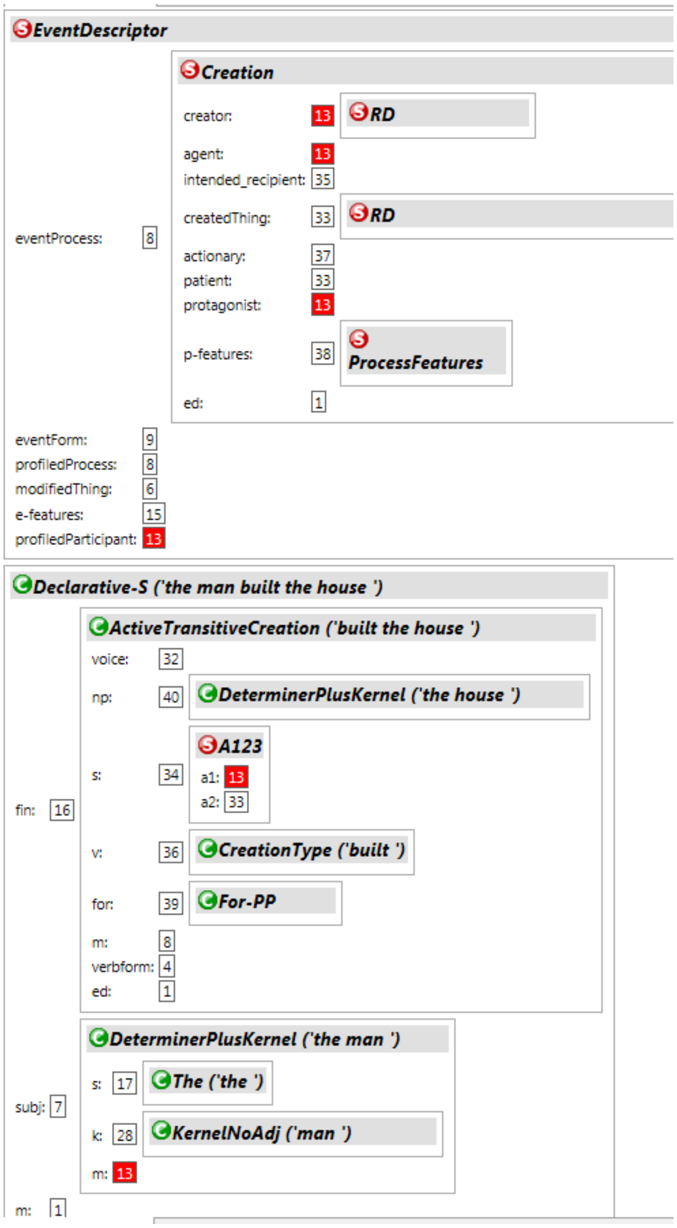
There are also commercial products for text analysis that can predict, e.g., the buying intentions of customers from written product reviews [22] or that can act as chatbots [23]. While this approach works for simple cases, it will probably not scale to complex commands for autonomous systems, due to the lack of a sophisticated constructional grammar that captures deep semantics.

III. Background

A crucial aspect of recognizing human intentions is understanding language. There is significant evidence that people understand language by carrying out mental simulations of the actions and events described in speech or text [6], [21], [17], [5]. Any natural language understanding system that aims for cognitive plausibility ought to reflect these findings. We have developed and implemented such a system for controlling and interacting with autonomous agents; its use for intention recognition in natural language will be described in Section IV.

A. Embodied Construction Grammar

Embodied Construction Grammar (ECG) [18] is a construction grammar that was developed by the Neural Theory of Language (NTL) group at ICSI. ECG is rooted in research from cognitive linguistics [19], and provides a precise formalism and technical notation to describe the grammar and meaning of a language. Like other construction grammars [37] [14], ECG consists of a network of rules outlining form and function pairings; meaning is represented with a lattice of embodied schemas, which are used by grammatical constructions to bind schema roles to grammatical constituents. For example, in the sentence *the man built the house*, the meaning would be **Creation**, with *the man* filling the **creator** role, and *the house* filling the **createdThing** role (see Fig. 1 for a computational analysis of the sentence, called a Semantic Specification; note that the highlighted boxes show roles that are "co-indexed" with the **creator** role).



EventDescriptor

Creation

creator: 13 RD

agent: 13

intended_recipient: 35

createdThing: 33 RD

eventProcess: 8

actionary: 37

patient: 33

protagonist: 13

p-features: 38 ProcessFeatures

ed: 1

eventForm: 9

profiledProcess: 8

modifiedThing: 6

e-features: 15

profiledParticipant: 13

Declarative-S ('the man built the house ')

ActiveTransitiveCreation ('built the house ')

voice: 32

np: 40 DeterminerPlusKernel ('the house ')

s: 34 A123 a1: 13 a2: 33

fin: 16

v: 36 CreationType ('built ')

for: 39 For-PP

m: 8

verbform: 4

ed: 1

DeterminerPlusKernel ('the man ')

s: 17 The ('the ')

subj: 7

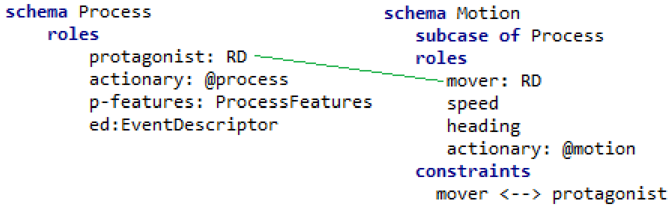
k: 28 KernelNoAdj ('man ')

m: 13

m: 1

Fig. 1: Partial SemSpec excerpt for ”the man built the house”.

Central to the theory and motivation behind ECG is that the schemas are embodied and cognitively plausible, with the schema inheritance lattice providing a conceptual network rooted in conceptual primitives like perception, causation, motion, and spatial relations, as depicted in Fig. 2. Thus, more



```
schema Process                              schema Motion
    roles                                      subcase of Process
        protagonist: RD                        roles
        actionary: @process                       mover: RD
        p-features: ProcessFeatures               speed
        ed:EventDescriptor                        heading
                                                  actionary: @motion
                                               constraints
                                                 mover <--> protagonist
```

Fig. 2: Schema lattice excerpt: ”Motion” is a ”Process”.

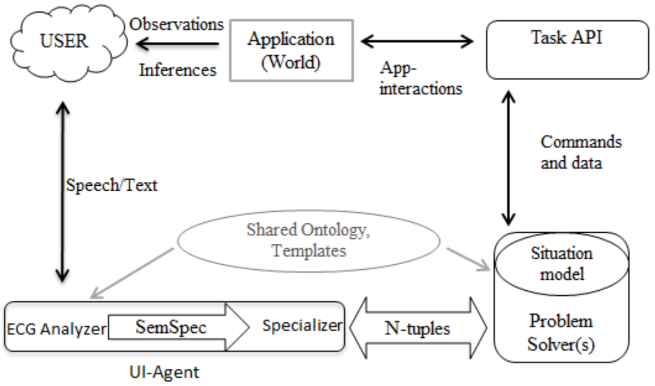


Fig. 3: System Architecture for Language Understanding.

”abstract” or specific schemas can be situated in a network of more embodied schemas; this is congruent with how previous work in linguistics suggests humans conceptualize world knowledge [19].

This combination of semantic and constructional compositionality allows ECG grammars to generalize across domains, and permits greater expressive power while maintaining cognitive plausibility. Previous work has discussed ECG's compositionality in more detail [16], but the key relevance here is that compositionality of both form and meaning prevents the combinatorial explosion that occurs with most grammars. ECG couples construction grammar's solution to scaling through constructional compositionality with its network of embodied schemas, which provide semantic compositionality. This is integral to our solution to the problem of indirect speech acts, as will be discussed in Section IV-B.

ECG has been computationally implemented. The ECG Analyzer [11], a probabilistic left-corner-parser, was also designed to be cognitively plausible, and reflects psycholinguistic studies on language processing. The Analyzer uses an ECG grammar to produce a best-fit semantic analysis, called a Semantic Specification (SemSpec) (see Fig. 1 and Fig. 6), of input text. The SemSpec is a rich data structure mapping information about the sentence's grammatical constructions to its meaning. This information is not itself a simulation of the events described in the input text; however, the SemSpec can be used to parameterize such a simulation [28] [29], or in the case of our current work, provide the front-end information for interacting with an autonomous agent.

B. System Architecture

We built an integrated system for natural language understanding and the control of autonomous agents for English, Spanish, and French [24], [44]. A video compilation of this system can be found at https://www.youtube.com/watch?v=mffl4-FqZaU. The system has since been extended to facilitate interaction with and between multiple agents [44], and is now being extended to novel domains, including computer games, metaphor, and mental spaces. The system's language analysis component can handle a wide range of grammatical phenomena, including conditionals (“if Robot1 had pushed the box, it would be in the room”), object control (“Robot1 caused the box to move into the room”), subject control (“Robot1 tried to push the box into the room”), and much more [43].

The system can be conceptually divided into a language-side and an action-side. The language-side (the left-hand portion of Fig. 3) is responsible for processing input speech or text, and producing a structured, semantic representation of that text, which is called an **n-tuple**. The action-side (the right-hand portion of Fig. 3) receives the n-tuple and uses it to carry out the task specified by the language. The nature of the task could range from performing an action (such as moving or modifying the world), responding to a query about the world state, or modeling a particular aspect of the world state based on an input assertion. The language and action sides, which can run on separate processes and machines, are bridged by n-tuples, which are transported between these agents using a network protocol.

The language-side consists of several subcomponents. The UI-Agent governs all interactions with both the user and the action-side, and is also responsible for controlling the flow of information from the text input to the n-tuple output. The text is first fed to the ECG Analyzer, which, as described in Section III-A above, produces a SemSpec. The Specializer then extracts task-relevant semantic information and formats it into a JSON array called an n-tuple, with the help of *n-tuple templates*, which declaratively instruct the Specializer on which information is necessary for the action-side. Additionally, n-tuple templates can provide *default* values for certain roles when deemed necessary by the underlying application, such as *speed* or *direction*. The Specializer also performs basic referent resolution for anaphora such as “it” or “one”, as in “the blue one”. One-anaphora resolution is a particularly difficult problem in computational linguistics, but our system is able to merge multiple descriptions to construct a more specific representation of a referent.

The action-side primarily consists of the Problem Solver, which unpacks an incoming n-tuple, determines what task is being asked of it, and how to carry out that task. The Problem Solver then makes API calls to the underlying application, and returns any relevant information to the UI-Agent (the answer to a query, or a description of any problems the Problem Solver ran into when carrying out the task). Additionally, the Problem Solver can request clarification on under-specified input; this feature is described in more detail in Section IV-A.

Each component of the system architecture depicted in Fig. 3 is also divided into **core** and **app** sides. The same core version of a component can be used across different domains and tasks. This facilitates relatively simple retargeting and integration with new domains, which involves building out the "app" or application-specific versions of the core components.

IV. Current Work: Methods for Intention Recognition

We have applied this system to the problem of recognizing a person's intentions from natural language. Two significant issues that surface are: 1) the ambiguity and under-specification of natural language, and 2) indirect speech acts, in which a person's intentions are implicit but not explicit. In this section, we discuss both problems, as well as the solutions we have implemented. Note that both solutions are general and application-independent; they have been applied to multiple robotics applications, and the same implementation could theoretically be used to communicate with other autonomous systems, such as self-driving cars, autonomous wheelchairs, or smart homes.

A. Clarification Dialog

Natural language is ambiguous, and the referent of a description is not always immediately apparent. There has been considerable work in Bayesian pragmatics about grounding intended references, using both contextual information and heuristics about communicative informativeness [20]. While these Bayesian models can be predictive, there are many HRI systems in which the margin of acceptable error is very low; in those systems, it is preferable to determine precisely what the human user intended. This can be done with clarification dialogs, as we illustrate with our examples (**D.1** – **D.7**).

Humans ask each other for clarification frequently. The problem with many clarification dialog systems, particularly Spoken Dialogue Systems, is that the automated questions are generic and untargeted [40]. A more naturalistic system, such as that described in [40], would ask targeted questions, usually about under-specified referents.

The following utterance is a potential command given by a human user of our system:

USER: Robot1, push the box north! (**D.1**)

If there is more than one box in the Problem Solver's world model, such a command is ambiguous - the solver has no way of determining which box the user is referring to. The system could make assumptions about the user's intentions, but this is not always desirable, particularly in a high-stakes environment.

Solution: we have implemented a clarification dialog system for under-specified referents. An important feature of our system architecture is the bidirectional communication between the UI-Agent and the Problem Solver (Fig. 3), which is facilitated by n-tuples. When a referent is under-specified, as in **(D.1)** above, the Problem Solver can use the information in the n-tuple to produce a targeted question, which it communicates back to the UI-Agent. Crucially, the Problem Solver also sends back the original n-tuple, with the under-specified entries tagged appropriately.

ProblemSolver: which box? (**D.2**)

USER: the red one (**D.3**)

The language-side then produces an n-tuple from the users new input ("the red one"), which is integrated with the previous, under-specified n-tuple. If the new n-tuple is still too vague (e.g. there are two red boxes), the process continues iteratively until successful. The Specializer's referent resolution capabilities are integral to the process of resolving the new input with the under-specified n-tuple, particularly in the case of one-anaphora (see Section III-B for more details).

ProblemSolver: which red box? (**D.4**)

USER: the one near the green box (**D.5**)

So far, our implementation includes only under-specified objects, such as those encoded in noun phrases. The Problem Solver's system for grounding referents from n-tuples is complex, so these descriptions range from *the blue box* to *the box that Robot2 pushed north*. Theoretically, however, the implementation could be extended to other cases of ambiguity, since the n-tuple is a structured set of key-value pairings. This could include the type of action specified, as well as the parameters of such an action, such as *speed*, *distance*, or *direction*. As mentioned in Section III-B, the n-tuple templates can also provide default values for parameters not explicitly filled by input language; these values are contingent on the underlying application.

B. Indirect Speech Acts

In typical human communication, a speaker communicates some utterance to a listener. The performance of this utterance frequently serves several purposes. J. L. Austin defines three "levels" of performative speech acts [1]:

1) Locutionary
2) Illocutionary
3) Perlocutionary

A locutionary act is the content or surface meaning of an utterance. For example, at face-value, the sentence "It is warm in here" is an assertion about the temperature of the room. An illocutionary act is the intention of the utterance; in this case, the speaker may intend that the listener open a window. Finally, a perlocutionary act is the actual effect of the utterance; for example, the listener might actually get up and open a window. In that case, the speaker's intention would be satisfied.

These classifications are relevant for a successful interaction between a human and a robot. When accomplished through language, HRI entails a congruence between the illocutionary and perlocutionary acts. In other words, the human says something, intending some sort of effect, and the robot carries out an action that achieves that effect. The precise nature of the desired effect could range from answering a question to altering the state of the world. In our system, the standard grammatical *moods* of direct speech acts are mapped onto fillers of an n-tuple value called *predicate-type*; different *predicate-types* also have different *return-types* by default (e.g. the expected value to be returned to the language-side, by the action-side). A *return-type* of “error-descriptor” means that the language-side expects information about whether the action-side was able to carry out the task, whether that be processing an assertion or executing a command. The notion of a *return-type* is essential to an implementation of different speech acts, because it, along with the *predicate-type*, represents the speaker’s intentions of the utterance.

1) Assertion → assertion → error-descriptor
2) Interrogative
   a) Yes/No Question → query → boolean
   b) WHICH-Question → query → instance-reference
   c) WHERE-Question → query → location-reference
   d) WHAT-Question → query → class-reference
3) Imperative → command → error-descriptor

Frequently, the locutionary act is congruent with the speaker’s intentions, as in the utterance below:

USER: Robot1, push the blue box 5 inches north! **(D.6)**

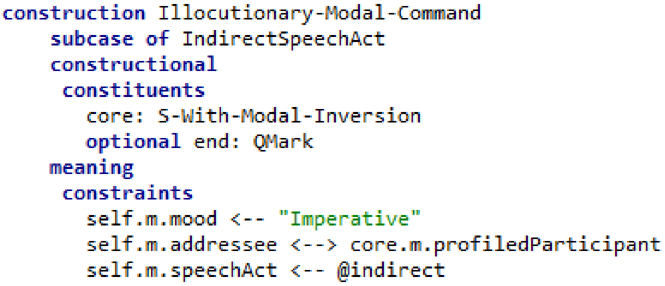
In these cases, the language-side of the system (described in Section III-B) would use the grammatical constructions present in the utterance to correctly identify the *mood* as an “Imperative”, and thus the *predicate-type* would be tagged as a “command”. The action-side then knows to treat the semantic content of the n-tuple as such.

It is also quite common for humans to express their intentions indirectly; these utterances are called *indirect speech acts* [33]. For example, the utterance could take a form typical to yes/no queries, but the speaker might actually intend for the listener to carry out a task:

USER: could you push the blue box 5 inches north? **(D.7)**

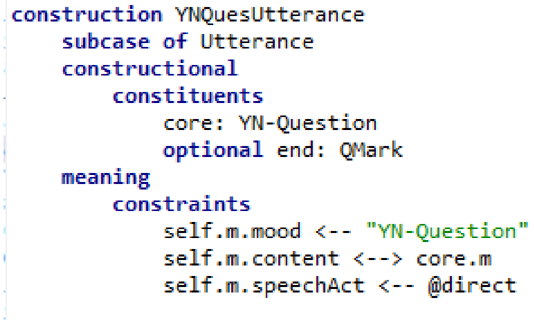
A typical language understanding system would interpret this utterance as a question about the system’s ability to push the box. However, a human listener would know that the speaker probably intends for them to treat the question as a request or command, and would act accordingly.

Solution: One way to handle such an utterance is to build rules about certain grammatical forms mapping onto certain discourse *moods*. While this might be difficult in some grammars, the compositional nature of ECG simplifies the process considerably. In this case, there is an entire class of indirect commands that take the constructional form of a yes/no question and begin with a *modal*, such as “could”, “can”, “would”, or “might”. To capture this, we wrote a construction to capture such illocutionary modal commands (Fig. 4). It includes an inverted modal yes/no question which recasts the mood of the speech act as an “Imperative”. We also mark the **speechAct** role as *@indirect*. ECG’s compositionality allows both the same core meaning and constructional patterns to be repurposed, so that the semantic information is still structured in the same way. This approach is quite robust, since it takes only a single construction to capture an entire class of common indirect speech acts.



```
construction Illocutionary-Modal-Command
    subcase of IndirectSpeechAct
    constructional
     constituents
       core: S-With-Modal-Inversion
       optional end: QMark
    meaning
     constraints
       self.m.mood <-- "Imperative"
       self.m.addressee <--> core.m.profiledParticipant
       self.m.speechAct <-- @indirect
```

Fig. 4: ECG construction for an inverted modal command.



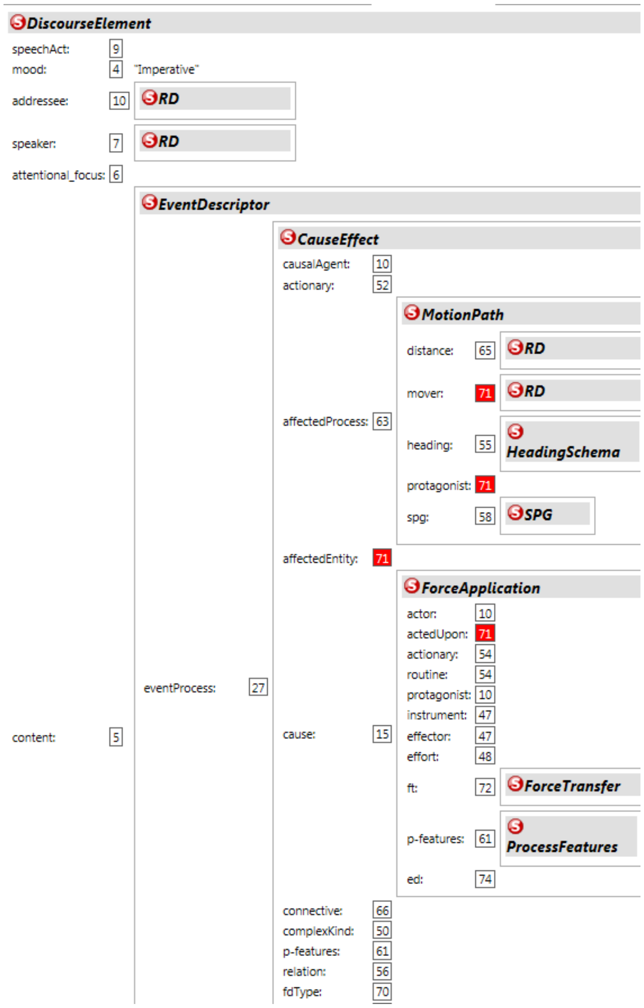
```
construction YNQuesUtterance
    subcase of Utterance
    constructional
        constituents
            core: YN-Question
            optional end: QMark
    meaning
        constraints
            self.m.mood <-- "YN-Question"
            self.m.content <--> core.m
            self.m.speechAct <-- @direct
```

Fig. 5: ECG construction for a yes/no question utterance (direct speech act).

With this new constructional rule, the Analyzer now produces two possible SemSpecs for the inverted modal utterance. The *command* interpretation is ranked higher than the *yes/no* interpretation, according to the Analyzer’s best-fit procedure, which uses the grammar’s rules and probabilities to assign “costs” to different analyses [11]. In this case, the *command* interpretation is ranked higher because the construction has a tighter constraint on the “core” constituent, requiring that it be of type “S-With-Modal-Inversion”. This constituent construction, which describes inversions like “could/would/might you push the box”, is much more specific than the general “YN-Question” construction, which is the required constituent of a *yes/no utterance* (see Fig. 5). The “YN-Question” construction includes modal inversions, but also other utterances like “is the blue box near the green one?” or “did Robot1 push the blue box?” Consequently, the *command* interpretation is a better fit, in terms of constituent specificity.

See Fig. 6 for an excerpt of the resulting *command* SemSpec; note that this excerpt omits the constructional (syntactic) tree. See Fig. 7 and Fig. 8 for a comparison of the constructional spans; the same structure appears compositionally below the high-level interpretation of the utterance as either a command or a question. This compositionality is a major strength of ECG.

In terms of the system architecture, the best-ranked SemSpec is then used to produce an n-tuple, which is sent to the



```
DiscourseElement
speechAct:          9
mood:               4  "Imperative"
addressee:          10  RD
speaker:            7   RD
attentional_focus:  6
content:            5
    EventDescriptor
    eventProcess:   27
        CauseEffect
        causalAgent:      10
        actionary:        52
        affectedProcess:  63
            MotionPath
            distance:    65  RD
            mover:       71  RD
            heading:     55  HeadingSchema
            protagonist: 71
            spg:         58  SPG
        affectedEntity:   71
        cause:            15
            ForceApplication
            actor:       10
            actedUpon:   71
            actionary:   54
            routine:     54
            protagonist: 10
            instrument:  47
            effector:    47
            effort:      48
            ft:          72  ForceTransfer
            p-features:  61  ProcessFeatures
            ed:          74
        connective:       66
        complexKind:      50
        p-features:       61
        relation:         56
        fdType:           70
```

Fig. 6: SemSpec excerpt for ”could you push the blue box 5 inches north?”.

Problem Solver. The Problem Solver assesses the *command* interpretation, and, if it makes sense according to the context model, executes the command. Of course, the n-tuple includes the fact that this interpretation is *@indirect*, so if necessary, the Problem Solver can request verification from the user that they did indeed intend a command, not a question.

In contrast, if a user states, “are you able to push the blue box 5 inches north?”, it is clear that a question is being asked, as opposed to an ambiguous command. In this case, the ECG Analyzer would only produce the *yes/no question* interpretation, because the sentence is not of the type “S-With-Modal-Inversion”. This exploits constructional patterns without requiring superfluous action on the part of the Problem Solver.

Linguists have also identified constructional patterns called Illocutionary Force Indicating Devices (IFIDs) [34], such as “I command that...” or “I promise that...”, which can be used to aid in classification of indirect speech acts. However, not all problems with indirect speech acts (or intention recognition) can be solved with constructional rules. Even in the case above, it is important that the SemSpec be marked as *@indirect*, so that the action-side is aware of the utterance’s categorization, and can act accordingly. In some cases, it might be appropriate for the Problem Solver to ask for more information from the user, with the aid of the clarification dialog system described in Section IV-A. Speech cues such as intonation offer additional pragmatic information, which we are currently exploring.

In general, ECG’s compositionality means that building a set of constructions to cover common indirect speech acts is actually a fairly constrained task. As observed with the modal command construction, incorporating these other constructions into ECG should not require any low-level changes, because this type of “discourse” information is at a higher level of abstraction. Of course, such models are not wholly comprehensive; for this broader coverage, we are interested in the approach described in [46], which integrated frameworks for evidence-based reasoning about uncertainty.

V. Conclusion

Any functional HRI application must be able to understand a person’s intentions, whether communicated through gesture, eye gaze, or language. We discuss an implemented system for natural language understanding, building on previous work in cognitive linguistics and embodied cognition [29], [24].

In this work, we focus on integrating linguistics research on intentions [20], [1], [33] to improve our existing system. We identify two key areas in which intention recognition was difficult for our language understanding system: 1) ambiguity or under-specification in natural language, and 2) indirect speech acts, in which an intention is implicit. For the first problem, we implement a clarification dialog, which allows the system to carry out a naturalistic interaction with the user and obtain more information. For the second problem, we build constructional rules to infer the correct utterance *mood* from indirect speech acts.

A. Future Work

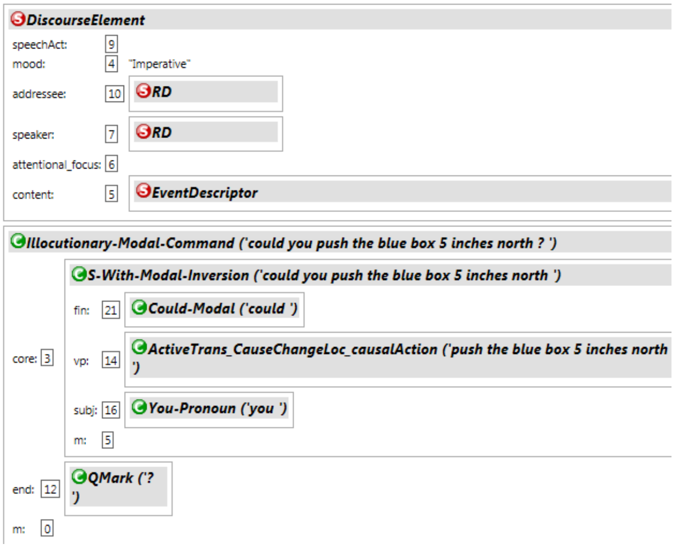
As mentioned in Section IV-A, the clarification dialog design could theoretically be extended to other parameters of an event, besides object specification. For example, a “Motion” event might involve *speed* and *distance*. If these parameters are not specified by the language, and no default values are provided for the application, the Problem Solver can tailor questions to that particular role. The n-tuple’s structured representation of events is ideally suited to this type of parameter-based clarification; a user’s response could then be integrated into the new n-tuple in a generalized way, since the unclear parameters are tagged as such. This would also be a powerful demonstration of the n-tuple paradigm.

In addition to using clarification dialog for other action parameters, we are considering an implementation that would request clarification when the intended effect of an utterance is unclear. Section IV-B points out that constructional rules are not always enough to infer the correct *mood* from a speech act; in cases of ambiguity, clarification dialog can be used to more precisely capture the user’s intentions.

We are also looking into using auditory information to inform inferences about intention and meaning. We have integrated the Kaldi speech recognition toolkit [30] into our system, and also plan to capture intonation information about an utterance, which can be incorporated into the ECG Analyzer's process for ranking SemSpecs. Intonation, along with other prosody information, can also help resolve other ambiguities that surface in text-only systems, such as formality, scope of negation, and questions.

Finally, just as auditory information could be incorporated into ECG, construction grammars could theoretically be used to capture other levels of meaning, from discourse to non-linguistic information like gestures [9]. Unlike ECG, these construction grammars have not been computationally implemented; here, an implementation would benefit substantially from a further review of current research in HRI covering gesture, eye gaze, and other non-linguistic data.

Appendix
SemSpec Constructional Tree Comparisons



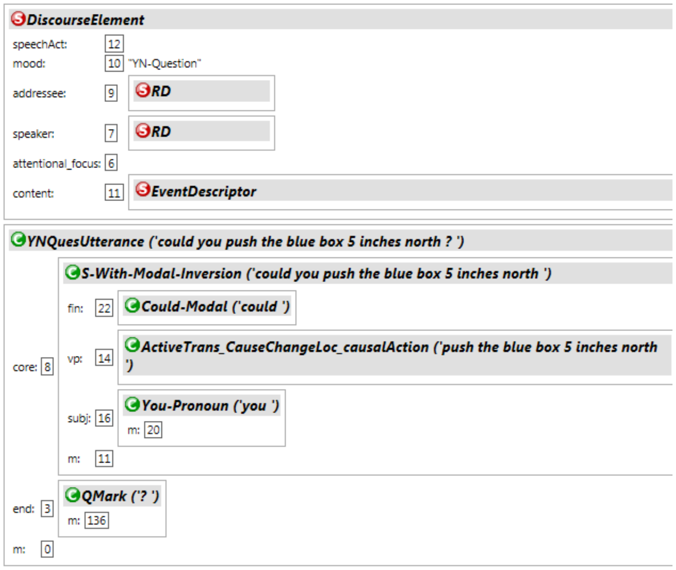
```
DiscourseElement
speechAct:         9
mood:              4  "Imperative"
addressee:         10  RD
speaker:           7   RD
attentional_focus: 6
content:           5   EventDescriptor

Illocutionary-Modal-Command ('could you push the blue box 5 inches north ? ')
         S-With-Modal-Inversion ('could you push the blue box 5 inches north ')
         fin:  21  Could-Modal ('could ')
core: 3  vp:   14  ActiveTrans_CauseChangeLoc_causalAction ('push the blue box 5 inches north
                   ')
         subj: 16  You-Pronoun ('you ')
         m:    5
end: 12  QMark ('?
         ')
m:   0
```

Fig. 7: Partial SemSpec for illocutionary command interpretation of "could you push the blue box 5 inches north?".

Acknowledgment

This work is supported by the Office of Naval Research grant number N000141110416 and a research grant from Google. Manfred Eppe received support from the German Academic Exchange Service (DAAD) via the FITweltweit program.



```
DiscourseElement
speechAct:         12
mood:              10  "YN-Question"
addressee:         9   RD
speaker:           7   RD
attentional_focus: 6
content:           11  EventDescriptor

YNQuesUtterance ('could you push the blue box 5 inches north ? ')
         S-With-Modal-Inversion ('could you push the blue box 5 inches north ')
         fin:  22  Could-Modal ('could ')
core: 8  vp:   14  ActiveTrans_CauseChangeLoc_causalAction ('push the blue box 5 inches north
                   ')
         subj: 16  You-Pronoun ('you ')
                   m: 20
         m:    11
end: 3   QMark ('? ')
         m: 136
m:   0
```

Fig. 8: Partial SemSpec for yes/no question interpretation of "could you push the blue box 5 inches north?".

References

[1] J.L. Austin. *How to do Things with Words*. Oxford: Clarendon Press, 1962.

[2] Daniel Paul Barrett, Scott Alan Bronikowski, Haonan Yu, and Jeffrey Mark Siskind. Robot Language Learning, Generation, and Comprehension. In *arxiv:1508.06161*, 2015.

[3] Emanuele Bastianelli, Giuseppe Castellucci, Danielo Croce, and Roberto Basili. Textual Inference and Meaning Representation in Human Robot Interaction. In *Joint Symposium on Semantic Processing*, 2013.

[4] Emanuele Bastianelli, Giuseppe Castellucci, Danilo Croce, Roberto Basili, and Daniele Nardi. Effective and Robust Natural Language Understanding for Human Robot Interaction. In *European Conference on Artificial Intelligence*, 2014.

[5] Ben Bergen. *Louder than Words: The New Science of how the Mind Makes Meaning*. Basic Books, 2012.

[6] Ben Bergen and Kathryn Wheeler. Sentence Understanding Engages Motor Processes. In *Proceedings of the Twenty-Seventh Annual Conference of the Cognitive Science Society*, 2005.

[7] Katrien Beuls, Remi Van Trijp, and Pieter Wellens. Diagnostics and repairs in fluid construction grammar. In *Language Grounding in Robots*. Springer, 2012.

[8] A. Bhargava, A. Celikyilmaz, D. Hakkani-Tur, and R. Sarikaya. Easy contextual intent prediction and slot detection. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8337–8341, 2013.

[9] Hans Boas and Ivan Sag. *Sign-Based Construction Grammar*. CSLI Lecture Note 193, Oct. 2012, 2012.

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

[11] John Edward Bryant. *Best-Fit Constructional Analysis*. PhD thesis, University of California at Berkeley, 2008.

[12] Rehj Cantrell, Matthias Scheutz, Paul Schermerhorn, and Xuan Wu. Robust spoken instruction understanding for HRI. In *International Conference on Human-Robot Interaction (HRI)*, 2010.

[13] Joyce Y Chai, Lanbo She, Rui Fang, Spencer Ottarson, Cody Littley, Changsong Liu, and Kenneth Hanson. Collaborative effort towards common ground in situated human-robot dialogue. In *International Conference on Human-Robot-Interaction (HRI)*, 2014.

[14] William Croft. *Radical Construction Grammar: Syntactic Theory in Typological Perspective*. Oxford University Press, 2001.

[15] Robin Deits, Stefanie Tellex, Pratiksha Thaker, Dimitar Simeonov, Thomas Kollar, and Nicholas Roy. Clarifying Commands with information-Theoretic Human-Robot Dialog. *Journal of Human-Robot Interaction*, 2012.

[16] Manfred Eppe, Sean Trott, and Jerome Feldman. Exploiting Deep Semantics and Compositionality of Natural Language for Human-Robot-Interaction. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems in Daejeon, Korea*, 2016.

[17] Jerome Feldman. *From molecule to metaphor: a neural theory of language*. MIT Press, 2006.

[18] Jerome Feldman, John Edward Bryant, and Ellen Dodge. A Neural Theory of Language and Embodied Construction Grammar. In *The Oxford Handbook of Computational Linguistics*, pages 38 – 111. Oxford University Press, 2009.

[19] Charles J Fillmore. Frame Semantics and the Nature of Language. In *Conference on the Origin and Development of Language and Speech*, volume 280, pages 20–32, 1976.

[20] Michael Frank and Noah Goodman. Predicting Pragmatic Reasoning in Language Games. In *Science, Volume 336, Issue 6084, pp. 998*, 2012.

[21] Arthur Glenberg and Michael Kaschak. Grounding Language in Action. In *Psychonomic Bulletin and Review 2002, 9 (3), 558-56*, 2002.

[22] Lexalytics Inc. Intention detection — lexalytics. https://www.lexalytics.com/technology/intentions, accessed 06/17/16.

[23] Wit.ai Inc. Wit – landing. https://wit.ai, accessed 06/17/16.

[24] Huda Khayrallah, Sean Trott, and Jerome Feldman. Natural Language For Human Robot Interaction. In *International Conference on Human-Robot Interaction (HRI)*, 2015.

[25] Geert-Jan Kruijff, Pierre Lison, Trevor Benjamin, Henrik Jacobsen, Hendrik Zender, and Ivana Kruijff-Korbayová. Situated dialogue processing for human-robot interaction. *Cognitive Systems*, 2010.

[26] Geert Jan Kruijff, Hendrik Zender, Patric Jensfelt, and Henrik Christensen. Situated dialogue and spatial organization: What, where... and why? *International Journal of Advanced Robotic Systems*, 2007.

[27] Cynthia Matuszek, Evan Herbst, Luke Zettlemoyer, and Dieter Fox. Learning to parse natural language commands to a robot control system. *Intl Symposium on Experimental Robotics (ISER)*, 2012.

[28] Srini Narayanan. *Knowledge-based Action representations for metaphor and aspect (KARMA)*. PhD thesis, University of California at Berkeley, 1997.

[29] Srini Narayanan. Mind changes : A simulation semantics account of counterfactuals. Technical report, University of California at Berkeley, 2012.

[30] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Nagendra Goel, Mirko Hannemann, Yanmin Qian, Petr Schwarz, and Georg Stemmer. The kaldi speech recognition toolkit. In *In IEEE 2011 workshop*, 2011.

[31] Matthias Scheutz, Gordon Briggs, Rehj Cantrell, Evan Krause, Tom Williams, and Richard Veale. Novel mechanisms for natural human-robot interactions in the diarc architecture. In *AAAI*, 2013.

[32] Matthias Scheutz and Kathleen Eberhard. Towards a framework for integrated natural language processing architectures for social robots. In *International Workshop on Natural Language Processing and Cognitive Science*, 2008.

[33] John Searle. *Speech Acts*. Cambridge University Press, 1969.

[34] John Searle and Daniel Vanderveken. *Foundations of Illocutionary Logic*. Cambridge University Press, 1985.

[35] Michael Spranger and Luc Steels. Co-Acquisition of Syntax and Semantics An Investigation in Spatial Language. In *International Joint Conference on Artificial Intelligence*, 2015.

[36] Mark Steedman. *The syntactic process*. MIT Press, 2000.

[37] Luc Steels. *Design Patterns in Fluid Construction Grammar*. John Benjamins Publishing, 2011.

[38] Luc Steels. *The Talking Heads Experiment*. Language Science Press, 2015.

[39] Luc Steels, Joachim De Beule, and Pieter Wellens. Fluid Construction Grammar on Real Robots. In *Language Grounding in Robotics*. Springer, 2012.

[40] Svetlana Stoyanchev, Alex Liu, and Julia Hirschberg. Towards Natural Clarification Questions in Dialog Systems. In *The Questions, discourse and dialogue symposium at AISB*, 2014.

[41] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *AAAI Conference on Artificial Intelligence*, 2011.

[42] Stefanie Tellex, Pratiksha Thaker, Robin L H Deits, Dimitar Simeonov, Thomas Kollar, and Nicholas Roy. Toward Information Theoretic Human-Robot Dialog. *Robotics: Science and Systems Conference*, 2012.

[43] Sean Trott. Core grammar: Content and partitioning. https://github.com/icsi-berkeley/ecg_grammars/wiki/Core-Grammar:-Content-and-Partitioning, accessed 06/20/16.

[44] Sean Trott, Aurélien Appriou, Jerome Feldman, and Adam Janin. Natural Language Understanding and Communication for Multi-Agent Systems. In *AAAI Fall Symposium*, pages 137–141, 2015.

[45] Matthew R Walter, Sachithra Hemachandra, Bianca Homberg, Stefanie Tellex, and Seth Teller. Learning Semantic Maps from Natural Language Descriptions. *Robotics Science and Systems*, pages 1–8, 2013.

[46] Tom Williams, Gordon Briggs, Brad Oosterveld, and Matthias Scheutz. Going beyond command- based instructions: Extending robotic natural language interaction capabilities. In *Proceedings of AAAI*, 2015.

[47] Tom Williams, Gordon Briggs, Bradley Oosterveld, and Matthias Scheutz. Going Beyond Literal Command-Based Instructions : Extending Robotic Natural Language Interaction Capabilities. In *AAAI*, 2015.