Learning Actions and Verbs from Situated Interactive Instruction for Embodied Cognitive Agents

Shiwali Mohan

April 11, 2013

Contents

1 Introduction 3

2 Situated Interactive Instruction 4

2.1 Properties of Instruction 4

2.1.1 Language and information content in instructions 4

2.1.2 Control over interactions and learning 5

2.1.3 Learning from examples 5

2.2 Agent Design Requirements 5

3 Related Work 7

3.1 Embodied Language Comprehension 7

3.2 Human Robot Dialog 7

3.3 Grounded Language Acquisition 8

3.4 Learning from Human-Agent Interaction 9

4 Preliminaries 9

4.1 Evaluation Domains 10

4.1.1 Perception 10

4.1.2 Actuation 11

4.1.3 Instructor Interface 11

4.2 Cognitive Architecture 11

4.2.1 Working Memory 11

4.2.2 Procedural Memory 12

4.2.3 Semantic Memory 12

4.2.4 Episodic Memory 12

4.2.5 Spatial Visual System 13

4.3 Problem Formulation 13

4.3.1 State 13

4.3.2 Actions 13

4.3.3 Instructions 14

5 Comprehending Instructions 15
- 5.1 Imperative Sentences . . . 15
  - 5.1.1 Ambiguities and Challenges . . . 16
- 5.2 Grounded Comprehension Model for Imperative Sentences . . . 17
  - 5.2.1 Immediacy of Situation . . . 17
  - 5.2.2 Referent Sources . . . 18
  - 5.2.3 Contexts . . . 19
- 5.3 Current Status . . . 20
  - 5.3.1 Analysis and Evaluation . . . 20
- 5.4 Future Work . . . 21
  - 5.4.1 Partial Observability . . . 22
  - 5.4.2 Hypothetical Situation . . . 22
  - 5.4.3 Corrective Instruction . . . 22

6 Learning Action Verbs from Instructions 22
- 6.1 Verbs and Actions . . . 23
  - 6.1.1 VerbNet . . . 23
  - 6.1.2 Action Goals . . . 24
  - 6.1.3 Task Execution . . . 25
- 6.2 Action-oriented Verb Representation . . . 26
  - 6.2.1 Map . . . 26
  - 6.2.2 Action Concept Networks . . . 27
  - 6.2.3 Behavior . . . 28
- 6.3 Acquisition . . . 29
  - 6.3.1 Preliminary Analysis . . . 30
- 6.4 Summary . . . 31
- 6.5 Future Directions . . . 32
  - 6.5.1 Structural-level Extensions . . . 32
  - 6.5.2 Learning from Hypothetical States . . . 33
  - 6.5.3 Robust Learning . . . 33

7 Timeline 33

A Model of Interaction 38
- A.1 Requirements . . . 38
- A.2 Formulation . . . 39
  - A.2.1 Event . . . 39
  - A.2.2 Segment . . . 39
  - A.2.3 Stack . . . 40

1 Introduction

As computational, robotic agents become pervasive in human society, the challenge of supporting complex intelligent behavior in novel environments is becoming increasingly important. The agents are expected to be long-living intelligent agents that demonstrate reasonably complex behaviors on a variety of novel tasks, operate with a certain degree of autonomy, and participate in collaborative human-agent efforts. In order to meet these requirements, the agents must be efficient online learners. They must actively acquire diverse types of knowledge including object recognition, semantic organization and categorization, spatial relations, and task execution from their experiences in their environment and exploit this immediately for performance. Learning through self-directed exploration alone in complex environments is challenging. The agent has to solve several complex problems related to discovering structure in sensory unlabeled data, generating and testing several hypotheses, performing deduction and inference, and solving structural and temporal credit assignment problems. This has motivated research on exploiting human-agent interaction to provide supervision and feedback to agent's learning mechanisms.

Prior research has looked at learning from different kinds of human-agent interaction. There has been extensive work on learning from demonstration such as early work on learning to fly a simulated aircraft (Sammut et al., 1992). Several of such works have focused on deriving action policies from *embodied* example traces of expert behavior. The demonstration trace for the instructor are obtained either through teleoperation or shadowing (Argall et al., 2009). These traces consists of relevant state-action pairs from which the agent derives behavior policies. Other works have focused on incorporating human-agent interaction as feedback on agent's performance in a reinforcement learning framework (Maclin and Shavlik, 1996; Thomaz et al., 2006). These lines of research have focused on *low-level* human-agent interactions that are close to agent's processing units (state, action, reward). Such approaches have been shown to be useful for learning control policies for various kinds of (*primitive*) behaviors such as manipulating objects, locomotion, and obstacle avoidance.

Learning more complex behaviors (or tasks) such as *setting up the table* can be characterized as instantiating primitive behaviors in accordance with their goals and composing these primitives into a policy that will achieve the goal. Learning complex behaviors from demonstration traces is a challenging problem, typically requiring a large number of long example sequences. Demonstration-based approaches also assume similarity of embodiment between the instructor and the agent achieved either by teleoperation or shadowing. An alternative (and complementary) approach is to learn from *knowledge-level* interactions that do not directly access agent's relevant processing units but exploit knowledge and perceptions shared between the human and the agent (*common ground*). The flexibility and the ease of use for humans gives natural language a strong advantage as a candidate modality for supporting knowledge-level interactions.

The focus of this work is to study learning from natural language, knowledge-level human-agent interactions. We focus on learning from a specific kind of interaction - *situated interactive instruction*. The human-agent interaction occurs within a real-world context. The agent grounds linguistic symbols in the interactions to objects, their properties, spatial relationships, and action definitions. Through this *grounding* process, the agent extracts specific examples from the real-world scenarios that form the basis of learning. The interaction is *mixed-initiative* and distributes the onus of learning among both participants. This thesis studies learning action[1] knowledge from interactive instructions. Through a sequence of interactions, the instructor leads the agent through an example sequence of an task. The agent effectively combines knowledge in interactions with its experience in the environment to extract generally applicable knowledge and adds it to its repertoire. Along with learning the execution knowledge corresponding

[1]This document uses the terms behavior, task, and action interchangeably.

to the novel action, the agent also acquires linguistic and semantic knowledge of verbs that aid in further communication about the newly learned task/action and are useful in learning complex, hierarchical action execution policies.

This thesis will focus to two key challenges in learning actions from situated interactive instruction.

- The first concerns grounded comprehension of language through which the agent translates information in linguistic interactions to its internal processing and reasoning units.
- The second addresses the issues related to knowledge representation for verbs in physical systems, and how this knowledge can be acquired from human-agent interaction.

2 Situated Interactive Instruction

The agent is embodied in a task-oriented domain and can communicate with the instructor through restricted natural language textual interface. To successfully acquire diverse knowledge from interactive instructional dialog, the agent must be endowed with several capabilities including natural language comprehension, dialog management, perception and action, and learning. The design of these capabilities and their integration is motivated by the structure of interaction that arises in learning with instruction and the information available to learn from. The following sections identify the properties of instructional interaction and derive requirements on agent design in accordance with the problem description and properties of human-agent interactions.

2.1 Properties of Instruction

Prior research on collaborative task execution dialog (Oviatt and Cohen, 1991; Rich and Sidner, 1998; Byron and Fosler-Lussier, 2005; Scheutz et al., 2011) asserts that such dialog is rich in information about the task. It may identify relevant components, task decompositional structure, goals, and constituent actions. Interactive instruction can be characterized as a special case of collaborative problem solving and task execution in which one of the participants (the instructor) possess the knowledge to do the task leads the other participant (the agent) through an example execution of the task. The agent combines the experience of performing a task with knowledge about the task in the human-agent dialog to acquire general task knowledge. Some properties of collaborative, task-oriented, communication that are relevant to this work are described below.

2.1.1 Language and information content in instructions

P1 **Referential Language**: Language is used to identify and bring to the attention of the collaborator, objects of interests, actions to be taken in the environment, useful relationships between objects, and feedback from the environment.
Examples:
"so you see <u>that thing on the wall on the right</u>" (Byron and Fosler-Lussier, 2005)
"You should be seeing a <u>door in front of you</u>" (Scheutz et al., 2011)
The referential nature of the language allows the listener to *situate* the communication in the current task and create beliefs about the speakers intentions and act accordingly. Natural language is ambiguous and often requires the hearer to exploit diverse contexts for comprehension. Some challenges related to common ambiguities are discussed in Section 6.

2.1.2 Control over interactions and learning

P2 **Mixed-Initiative**: The initiative of the communication is distributed among the participants of the interaction. The instructor takes initiative in structuring and decomposing the task, identifying useful objects and relationships, and suggesting the appropriate action to take in the current situation. The learner takes initiative in resolving ambiguities in the instructions, and guides its learning by asking for instructions in situations where it lacks knowledge to make progress in the task. This offers an opportunity for mixed-control of learning. Often in human controlled interactive learning such as learning by demonstration, the onus is on the instructor to provide good examples from the feature space so that the agent can acquire general hypotheses. In contrast, with interactive instruction, the instructor can rely on the agent to initiate an interaction when needed. This approach can speed instruction by eliminating the need for the instructor to carefully structure the interaction or repeatedly verify agent's learning.

2.1.3 Learning from examples

P3 **Incremental Elicitation**: The instructions rarely contain the complete procedure to do a task in a single interaction. Rather the dialog unfolds as the listener acts in the environment in response to the instructions, resulting in complex interactions. For example, an instructional dialog for following a particular path in the environment is below. Such interaction requires that sensing, language understanding, and behavior must be intertwined and must be performed online. The information provided for performing the task is spread out in time. To learn from the interaction, the agent must maintain the memory of task performance that is available for later analysis and generalization.

```
Instructor: OK, continue to walk straight.
Robot (continuing straight): OK.
Instructor: You should be seeing a door in front of you.
Robot (looking out for a door): Yes.
Instructor: Good, go through that door.
Robot (moving through the door): OK, I'm through the door.
Instructor: Alright. Keep going. There should be a whiteboard.
Robot (looking for whiteboard): OK, I'm not seeing it yet. There it is.
Instructor: Great, then you should see an intersection, go there.
Robot (looking out for an intersection while moving): Got it,OK.
```

P4 **Situation Specificity**: Instructions are highly specific to the current situation (observable to the speaker and the listener) and do not describe how a task can be executed in general. Example:
"Good, go through that door." (Scheutz et al., 2011)
"And put it so that it's covering the hole in the bottom of that little cap." (Oviatt and Cohen, 1991)

P5 **Sparse Data**: The human involvement severely limits how much data is available to learn from. Repeated similar interactions with the human are undesirable. A human instructor provides few, highly situation specific examples of knowledge to acquire general task knowledge.

2.2 Agent Design Requirements

Based on the properties of the human-agent interaction in instructional settings and the goals of this thesis, we derive the following requirements for designing agents that can learn from instructional human-agent interaction. There are following three classes of computational challenges that have to be addressed in designing instructable agents.

- R1 **Referential Comprehension**: To comprehend linguistic instructions, the agent must transform the information in natural language to its internal representations (P1). Human generated instructions can be linguistically complex and difficult to interpret. Even linguistically simple instructions can be ambiguous, requiring the agent to exploit context and domain knowledge to generate completely specified interpretation. Section 5 discusses the important challenges of referential comprehension and presents an approach to addressing these challenges.

- R2 **Integrative Interaction**: The agent is continually embedded in an environment with an instructor who communicates with the agent. The agent must comprehend the instructors commands, comprehend the scene, reason about the possible ways to execute the command in the scene, act, and respond to the instructor (P3). This imposes constrains on the design of the interaction system. It must be,

  - R2a **Integrative**. The model of interaction should serve to integrate various cognitive capabilities encoded in the agent, allowing the agent to plan over and reason about a combined space of linguistic processing, behavior, and learning.
  - R2b **Flexible**. The model should allow both participants of the conversation to change the focus of communication (*flexible initiation*) regarding various aspects (perceptual, spatial, semantic, procedural) of task (*flexible content*). Furthermore, the model should not impose any requirements on the order in which task-relevant knowledge is presented (*flexible command*).
  - R2c **Task-oriented**. The model should capture the structure of task oriented communication and provide discourse context for resolving ambiguities, correct interpretation of instructions, and organize dialog so that it is useful in task execution and learning.

  Studying human-agent interaction is not the immediate focus of this thesis. However, since an interaction system provides the framework for learning with interaction, we have implemented an basic interaction model that captures the useful structure of task-oriented discourse. The model is adapted from the attentional theory of discourse (Grosz and Sidner, 1986; Rich and Sidner, 1998) which has been extended to support learning from interactions. The implementation details are in Appendix A.

- R3 **Incremental Learning**: The experience of interactive execution of novel tasks provides the agent with different kinds of information that is useful for acquiring knowledge about task execution and communication. Agent's learning should be

  - R3a **Diverse**. To learn novel actions and related verbs to communicate about these actions, the agent must acquire knowledge in diverse categories. The agent must acquire linguistic knowledge to map the linguistic form (verb phrases, prepositions, and noun phrases) in action commands to instantiated actions. The agent also acquires knowledge of how to execute the action including the availability and termination condition.
  - R3b **Assimilative**. The acquired knowledge should integrate seamlessly with prior knowledge that either has be pre-encoded in the agent or has been acquired by the agent through other experience with the environment. The knowledge of verbs should assimilate with perceptual knowledge of objects and knowledge of spatial relationships.

R3c **General**. Through instructional interaction, the instructor provides situation specific (P4) example executions of a new task. The agent must extract generally applicable linguistic, semantic, and procedural knowledge from the instruction sequence and the corresponding observed changes in environment. Instructional learning limits the number of examples the agent has to learn from (P5), requiring the agent to exploit its domain knowledge heavily to achieve general learning.

R3d **Comprehensive**. The learning mechanism should allow the agent to learn a variety of English action verbs through interactive instruction.

R3e **Active**. The agent exploits all its knowledge to interpret and execute instructor's command. It can initiate communication if it fails at any stage of processing (P2). The agent should actively seek examples of concepts that are hard to learn and avoid asking for multiple examples of easily acquired concepts.

Section 6 discusses the acquisition of diverse knowledge for actions in further detail.

3 Related Work

In light of the goals of the thesis, characteristics of interactive instruction, and the requirements they impose on agent design, we briefly and broadly discuss the research areas that are related to our work on learning with interactive instruction. The research work in the following communities have focussed on subsets of requirements discussed above.

3.1 Embodied Language Comprehension

SHRDLU (Winograd, 1972) was an early natural language understanding system that carried out conversations about a virtual blocks world. The robot could follow simple object manipulation action commands (R1a). A recent work (Goertzel et al., 2010) describes software architecture which enables a virtual agent in an online virtual world to carry out simple English language instructions grounded in its perceptions and actions (R1a). This system uses knowledge from external sources such as FrameNet and other similar sources associate semantic meaning to linguistic utterances. These systems are limited in the number of instructions they can follow as they do not learn from their experience with manipulating the environment (R3). The virtual nature of the environment simplifies the challenge of referential comprehension (R1).

3.2 Human Robot Dialog

Research in human-robot interaction has primarily focussed on the integrative interaction requirement (R2) to develop systems that maintain a multi-modal communication with a human partner and solve collaborative tasks. Rich and Sidner (1998) demonstrated an application-independent collaboration manager that allows an agent to provide intelligent, mixed-initiative assistance for air-travel application. The agent is a planning system that interacts with a human user to determine the constraints and goals of their air-travel and manages reservation. The system demonstrates limited, short-term acquisition of user-specific knowledge that is useful for generating the required plan. It, however, does not learn new tasks (R3).

Cantrell et al. (2010) demonstrate a natural language understanding architecture for human-robot interaction that integrates speech recognition, incremental parsing, incremental semantic analysis and situated reference resolution. The semantic interpretation of sentences is based on lambda representations and combinatorial categorial grammar. Cantrell et al. (2011) extended the system to learn action verbs from linguistic instruction. The robot is able to understand human speech and can learn verbs and actions from definitions. A sample interaction is below.

```
H: I want you to follow me.
R: I don't know how to follow.
H: it means that you should stay within 1 m. of me.
R: okay.
```

Using lambda-calculus, the agent is able to ground the linguistic definition of the action verb *follow* into its action primitives. This knowledge is immediately available for execution. The method relies on the instructor to provide definitions of verbs. Such representations are useful in defining simple behavior policies, but cannot express complex policies that implement situation specific execution. This learning mechanism also does not deal with incremental elicitation (P3) of instruction. The work also does not comment on how the mechanisms can be extended to demonstrate general learning (R3), to assimilate new knowledge with prior knowledge (R3b), or to learn concepts other than action verbs (R3a).

3.3 Grounded Language Acquisition

Computer Vision community has studied the problem of grounding different linguistic components. There has been extensive research on grounded acquisition of nouns from labeled pictures (Barnard et al., 2003; Gupta and Davis, 2008) and computer-generated visual scenes (Roy, 2002), associating linguistic descriptions with relationships (Kollar et al., 2010) and detect verbs in visual perception through activity recognition (Siskind, 2001). Typically, this line of work focuses on detecting regularities and correlation in linguistic and visual input (R1a, R3c) addressing the referential natural of language (P1). These methods limited to classification and labeling tasks, and do not make any claims about how learning can be assimilated with knowledge acquired through other learning mechanisms (R3b).

A significant line of research in grounded language acquisition investigates grounded interpretation of linguistic route instructions. Understanding route instructions involves grounding objects and landmarks, spatial relationships, and transforming the linguistic instruction to an formal plan that can be executed in the environment. MacMahon et al. (2006) assume a linguistic model of language and spatial navigational instructions and learn the sequence of implicit and explicit action implied by the action command from a corpus of English instructions. Other works acquire a joint model of linguistic instruction and action plans (Chen and Mooney, 2011) or perception (Matuszek et al., 2012) from annotated corpus of English instruction. These methods are robust to free-form natural language that might be generated by humans. Although these methods provide solutions for the referential nature (R3a) of language, it is unclear how these methods can be extended to demonstrate diverse, comprehensive learning (R3a, R3b, R3c). These methods are also silent on how these mechanisms can be implemented in interactive agents.

Learning grounded representations of words has been previously studied in the context of learning to describe a visual scene (Roy, 2002), understanding descriptions of scene (Gorniak and Roy, 2004), understanding spatial directions (Kollar et al., 2010), and understanding natural language commands for navigation Tellex et al. (2011). These works exploit the hierarchical, compositional structure of language and acquire associations between words and their referents in the physical world. They have focused on batch learning from free-form English corpora (and corresponding visual scenes/plans) generated by human viewers/users. This has led to development of mechanisms that are robust to user errors and imperfect language use. In general, these methods acquire diverse knowledge (R3a, R3b, R3c) from examples presented in the corpus of instructions that allows them to generate grounded representations of novel instructions

(R1a). They address the referential nature (R1) of instructions. However, generating an annotated corpus is expensive. The agents are prone to failure if their training is insufficient for grounding a novel instruction. An interactive agent on the other hand will robust switch to learning mode if it is unable to comprehend the instruction. It is unclear if data extensive, corpus-based, batch learning mechanisms can be effectively incorporated in online and incremental human-agent interactions (P2), allowing the agent to guide its own learning (R3e).

3.4 Learning from Human-Agent Interaction

Human-agent interaction for learning with instruction can be viewed on a continuum of instructor/agent control. Systems that exploit learning from examples, demonstration, or imitation (Sammut et al., 1992; Van Lent and Laird, 1999; Dinerstein et al., 2007) place the control of instruction with the instructor. The agent observes a human perform the task, maps that performance onto it own capabilities and tries to induce knowledge, goals, or reward function that the human had that produced the behavior. These interactions are low-level interactions in which the agent and the instructor either share the state-action space, or completely map the others state-action spaces to their own. The other extreme of the continuum, where the agent controls almost all aspects of learning. The instructor is limited to providing low-level feedback on agent performance, usually within a reinforcement learning framework (Maclin and Shavlik, 1996; Thomaz et al., 2006; Knox and Stone, 2010).

Early work done in learning by instruction by Huffman and Laird (1995) demonstrate a learning system that focuses on agent-initiated interaction, where instruction is directed by impasses arising in a Soar agent. The learning is completely controlled by the agent (R3e) and is focussed on acquisition of procedural knowledge for task execution (R3c). Our work extends this design to incorporate referential comprehension (R2) and integrative interaction (R1). We also aim at comprehensive learning of different kinds of tasks and goals in complex robotic domains.

Chen et al. (2010) describe a unified agent architecture for human-robot collaboration that combines natural language processing and common sense reasoning. They developed a planning agent that relies on communication with the human to acquire further information about underspecified tasks. The agent also demonstrates limited learning by acquiring novel common sense rules through dialog. Although the work demonstrates integrated (R1) natural language processing, common sense reasoning, planning, and robot navigation and manipulation, it is silent on how their design could be extended to demonstrated diverse and comprehensive learning (R3a, R3b, R3c).

Recent work by Allen et al. (2007) describes a collaborative task learning agent that learns executable task models from a single collaborative session (R2a, R2c) of demonstration, learning, and dialog. The human teacher provides a set of tutorial instructions accompanied with related demonstrations in a shared environment, using which the agent effectively acquires task models (R3c). The initiative of dialog is on the human user, however the agent controls certain aspects of its learning by making generalizations about certain tasks without requiring the human to provide a large number of examples (R2b).

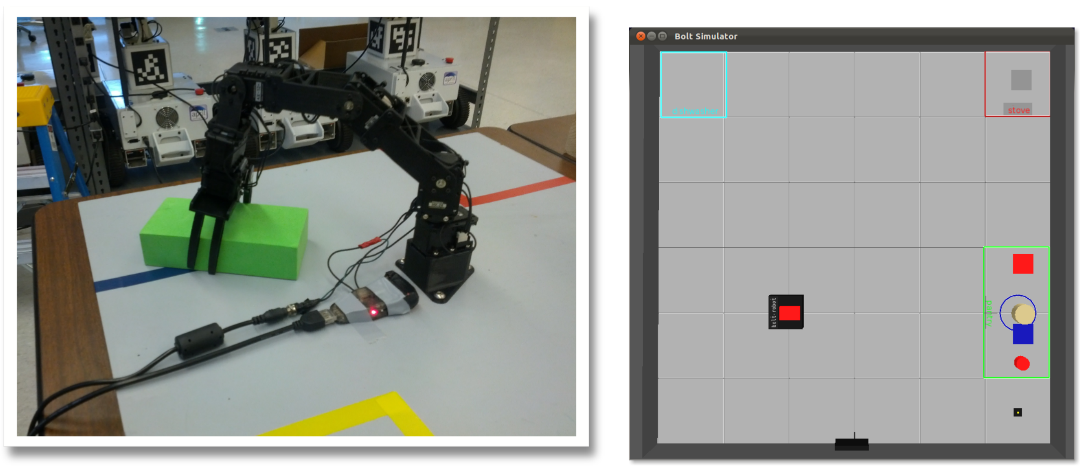
4 Preliminaries

This research requires integration of several different cognitive capabilities in a single agent. To focus this research, the following real-world and near-real-world environments will be used to motivate research and evaluation of agent designs. The embodiment of agent in rich, complex domains brings forth interesting questions related to language comprehension and learning with instruction.

4.1 Evaluation Domains

- **Table-top Robot**: The agent exists in a simple table-top environment (in Figure 1a) with a robotic arm and a Kinect sensor. The environment is composed of two kinds of objects: foam blocks of different colors, sizes, and locations that can be manipulated by the agent using the robotic arm and specific areas on the table that have functional properties. The blocks and the locations are perceived through the Kinect sensor. We assume the environment is completely observable.

- **Kitchen**: The kitchen domain (in Figure 1b) is a complex (simulation only) variant of the table-top environment described above. The agent in this domain move about and visit different locations. The objects are placed at different locations in the environment. The sensing in this is distance limited, the agent can only perceive objects within a certain radius. Partial observability of this domain motivates the use of long-term memory of agent's experience for linguistic comprehension.



(a) (b)

Figure 1: Experimental Domains

4.1.1 Perception

The perception system generates an object-oriented representation from the continuous data stream (3D point cloud) available from the Kinect. It first segments the scene using standard segmentation schemes available in the computer vision literature. Each object can is associated with a vector of perceptual symbols (*red*, *large*, *triangle*) corresponding to the three perceptual properties - *color*, *size*, and *shape* available through the camera. The perceptual symbols are extracted through K-Nearest Neighbor(KNN) classifiers with Gaussian weightings built for the perceptual properties. As an example, a perceptual symbol `R43` is associated with the region in the color feature space that corresponds to the word *red*. These symbols, along with position and bounding box information, are used to create a symbolic representation of the object, which is provided to the agent. The perceptual classifiers are trained through instruction.

4.1.2 Actuation

To act in the world, the agent sends discrete commands to the robot controller. Following classes of actions are available to the cognitive agent (constrained by availability in the domain).

- *Object manipulation* such as `pick-up (object-id)`, `put-down (x,y,z)` allows the agent to move objects on the work area.
- *Locomotive* commands such as `goto(x,y,z)` move the robot to the specified co-ordinates.
- *Functional manipulation* commands such as `turn-on(stove)` change the functional state of the locations.

The low-level behaviors corresponding to these action commands are pre-programmed in the robot controller and the environment. The environment provides a degree of feedback corresponding to successful, failure, or ongoing status of these action commands facilitating a closed-loop control.

4.1.3 Instructor Interface

The human-agent communication is facilitated by a chat interface. Messages to the instructor are parsed by the Link-Grammar parser (Sleator and Temperley, 1991) to extract part of speech tags and sentences structure. The structural representation of a sentence is then provided to the agent for further processing and comprehension. The messages from the agent are converted to language using pre-defined templates. The instructor can also an object by clicking on it in a live camera feed, and the selection is made known to the agent.

4.2 Cognitive Architecture

The Soar cognitive architecture (Laird, 2012) has been applied to a wide variety of domains and tasks including natural language understanding and robot control. Recent extensions to Soar, including episodic and semantic memories, as well as visual-spatial reasoning systems make Soar a suitable candidate architecture for designing agents that learn with human-agent instructional interaction as well as investigating several interesting aspects of agent-oriented natural language processing.

4.2.1 Working Memory

Working memory maintains symbolic relational representations of current and recent sensory data, current goals, and the agent's interpretation of the current situation. Thus, working memory holds the current state and operator and is Soar's *short-term* knowledge, reflecting the current knowledge of the world and the status in problem solving. Working memory buffers provide interfaces to Soar's long-term memories, the perceptual system, the robot controller, and the instructor interface.

The perception system in our agent segments the scene in to objects and locations and associates symbolic feature vectors with them. This information is represented in the working memory (R1a). Apart from the representation of the environment, the working memory holds the state of human-agent interaction and the current focus of conversation and information retrieved from long-term memories.

4.2.2 Procedural Memory

Procedural memory contains Soar's knowledge of how to select and perform discrete actions (called operators), encoded as if-then rules called productions. Productions fire in parallel whenever they match working memory and support the proposal, evaluation, selection, and application of operators. Operators are the locus of decision making in Soar. Once an operator is selected, rules sensitive to its selection and the current context perform its actions (both internal and external) by modifying working memory. Whenever procedural knowledge is incomplete or in conflict for selecting or applying an operator, an impasse occurs and a substate is created in which more reasoning can occur, including task decomposition, planning, and search methods. In Soar, complex behavior arises not from complex, preprogrammed plans or sequential procedural knowledge, but instead from the interplay of the agent's knowledge (or lack thereof) and the dynamics of the environment.

Soar acquires new procedural knowledge through *chunking*. Chunking creates rules from reasoning occurring in the substate. When a result is created in substate, a rule is compiled. The conditions of the this rule encode the working memory elements that existed before the substate and were necessary for creating the result. The rule is added to the procedural memory and is immediately available. This learning mechanism allows the agent to perform explanation based learning (R3a) of general action execution rules using an example execution of an action and pre-encoded domain models.

Procedural memory in our agent encodes knowledge of domain-specific lexical processing, mixed-initiative human-agent interaction management, grounded comprehension, and primitive action execution (R1c) and corresponding domain models (R3a). As the agent acquires knowledge for executing novel tasks, it is added to agent's procedural memory as new rules.

4.2.3 Semantic Memory

Semantic memory stores context'independent declarative facts about the environment. The agent can deliberately store working memory elements into semantic memory and it can retrieve them by creating a cue in a working memory buffer. The best match to the cue (biased by recency and frequency) is retrieved from semantic memory to working memory.

In our agent, semantic memory encodes diverse kinds of knowledge (R3a) including syntactic, surface-level realizations of verbs along with declarative structures representing the goal of the action corresponding to the verb (requirement R1c). It also encodes the relationship of verbs with situated representations of (acquired and preprogrammed) prepositions and objects (R3b).

4.2.4 Episodic Memory

Episodic memory stores context-dependent records of the agent's experiences. It takes snapshots of working memory (episodes) and stores them in chronological order, enabling the agent to retrieve both the context and temporal relations of past experiences. The agent can deliberately retrieve an episode by creating a cue in a working memory buffer. The best partial match (biased by recency) is retrieved and added to working memory. Episodic memory facilitates acquisition of action-execution knowledge through retrospective forward projection by automatically encoding all interactions accompanied by changes in sensory perception. The agent can review past instructions and observe the resulting changes in the environment and its own internal state. This allows the agent to compile information distributed temporally in interactions into procedural knowledge of the novel task (P3). Episodic memory is also useful in situated comprehension for resolving referents to shared experiences in the environment (R1b).

4.2.5 Spatial Visual System

Soar contains a task-independent Spatial Visual System (SVS) that supports translations between the continuous representations required for perception and the symbolic, relational representations in Soar. The continuous environment state is represented in SVS as a scene graph composed of discrete objects and their continuous properties. Binary spatial predicates are computed when an agent issues a query for a specific predicate such as `X-axis-aligned(A,B)`. The set of predicates is task independent and fixed, but predicate extraction is controlled using task-specific knowledge. The agent acquires useful compositions of spatial primitives through instruction. References to prepositions are resolved to these compositions (R1a).

4.3 Problem Formulation

We formally construct the problem of learning actions with situated instruction as follows.

4.3.1 State

The agent maintains an *object-oriented* representation of the world. The current state of the agent is composed of a set of beliefs about observable objects. An object in the agent's view can be described as a set of symbolic[2] perceptual (*color*, *shape*, *size*) features and their value assignments. Formally, let $F = \{f_1, f_2, ..., f_n\}$ be the set of attributes observable in the environment. Every attribute f_i (such as *color*) has an associated domain $Dom(f_i)$ (such as {`r2`, `b11`, ...}) of symbols. An object $o \in O$ can be described as the set of attribute-value pairs, $o = \{(f, val_f)|f \in F, val_f \in Dom(f)\}$ that are currently known to the agent. In an earlier work (Mohan et al., 2012a), we have shown that domains of these attributes can be acquired through human-agent interaction, if the agent begins with incomplete knowledge about perceptual attributes. In this work, we assume that the domains of all perceptual attributes are known to the agent. Object category predicates (P_c) were defined that categorized objects in two classes, $location(o)$, and $block(o)$. For locations, state predicates (P_s such as *open*(o), *on*(o)) are also elaborated from the attributes.

Useful binary spatial relationships are encoded as *compositions* (C) of *directional* and *distance* primitives extracted from SVS. Directional primitives describe how two objects are aligned along each axis in 3D coordinate system. Distance primitives encode distances between two objects. The primitives are composed as conjunctions and disjunctions of directional primitives, and as distributions of distance primitives. Binary predicates ($p_{sp} \in P_{sp}$) corresponding to spatial relationships known to the agent are instantiated for objects on the scene ($p_{sp} = c(o_1, o_2)|o_1, o_2 \in O, c \in C$). In this work we assume that relevant compositions are known to the agent. The earlier work describes how the compositions corresponding to spatial prepositions such as *left-of*, *in* can be acquired from human-agent interaction.

The current state of the agent is described as all the unary and binary predicates that are currently true.

$$S = \{p \;|p \in P_c \cup P_s \cup P_{sp} \wedge p = true\}$$

4.3.2 Actions

We define two categories of actions, *primitive* PA and *complex* CA. Primitive actions are pre-encoded in the agent and complex actions are acquired through human-agent interactions. An action $a \in A =$

[2] Perceptual symbols such as `R24` (for the color red) are mapped to classes in the KNN classifier in the perception system and encode non-symbolic knowledge.

$PA \cup CA$ is described by:

- **parameters**, a set of objects ($O^a \subset O$) that are involved in the application of an action a.
- **availability conditions**, a set of state predicates ($P^a_{avail} \subset P_c \cup P_s$) that have to be true for the action to be available for application. These conditions are pre-encoded in operator proposal rules for primitives and are acquired for complex actions through experience in the environment and human-agent interaction.
- **action space** $A^a_c \subset A$, a set of *child* actions of a complex action, a. It is acquired through experience of performing an action in the environment. For $a \in PA$, $A^a_c = \phi$.
- **policy** $\pi^a : S \rightarrow A^a_c$ that selects child actions based on the state of the environment. Primitive actions do not have a selection policy. To apply a primitive action, the agent send commands to the robot controller that translates the discrete commands to closed-loop continuous control. The selection policy for complex actions selects primitive and complex actions known to the agent and is acquired through learning.
- **termination conditions**, a set of state predicates ($P^a_{term} \subset P_c \cup P_s \cup P_{sp}$) whose truth values signify that the goal of the action has been achieved and the action should be terminated. These conditions are acquired through human-agent interactions for complex actions.
- **model**, is encoded as the transition function $T^a(s'|s, p \in A^a_c) : S \times A^a_c \times S \rightarrow [0, 1]$ and it defines how states transition in response to action a in the environment. These models are useful in acquire policies for novel complex actions. They are pre-encoded for primitive actions. Transition functions of a complex action (a) is implicitly derived by simulating the relevant policy (π^a) over the current state (s) to obtain the next state (s').

In Soar, the knowledge of an action is encapsulated in an action *operator*.

4.3.3 Instructions

The instructions (I) are generated from a restricted grammar and a restricted vocabulary. In this work, we use two kinds of instructions, *action commands* or imperative sentences (I_{AC}) and *goal definitions* (I_{GD}). We define a set of grounding functions ($G = G_{action} \cup G_{goal}$) that translate instructions to the agent's processing units (actions, states)

$$G^K_{action}(i) : I_{AC} \rightarrow A$$

and

$$G^K_{goal}(i) : I_{GD} \rightarrow S$$

where K is the agent's knowledge that associates words in natural language to perceptual symbols, spatial compositions, and action operators. It includes lexical, syntactical, and *mapping* knowledge that connects linguistic items to physical and conceptual entities known to the agent. We assume that lexical and syntactical knowledge an pre-encoded in the agent. The agent acquires *mapping* knowledge. For clarity in this thesis, we assume that the agent begins with mapping knowledge of nouns, adjectives, prepositions, and primitive verbs. As the agent learns novel verbs and corresponding actions, it expands this knowledge set. Consequently, it is able to translate instruction composed with novel verbs to actions. Earlier work has established that mapping knowledge for nouns/adjectives, prepositions, and verbs can be acquired through instruction. Section 5 identifies different kinds of knowledge that can be used in grounding functions. Section 6 describes how a subset of this knowledge can be acquired through instruction.

5 Comprehending Instructions

Learning from natural language instruction involves situated comprehension - constructing a grounded representation of what is being said and how it relates to the current situation. Situated comprehension is one of the two major questions addressed by this thesis. The results of comprehension of a sentence are influenced by the intentions behind that utterance. By uttering an imperative sentence such as *"put that book in the shelf"*, the speaker intends the listener to perform an action. Comprehension of an imperative sentence may result in an action by the listener that achieves the intended goal. However, comprehension of utterances such as assertions - *"there is a blue couch in the living room"* results in the establishment of shared belief. Interpretation of an interrogation - *"where is the blue couch?"* results in a speech act that provides the requested information. Other utterances might result in perceptual simulation. Recognizing the intention behind an utterance is a significantly complex challenge and an open area of research. For the purposes of this thesis, we will only study imperative sentences and assume that the instructor utters these sentences with an intention for the agent to perform these actions in the environment. In the following sections, we discuss the challenges (in Section 5.1) in generating grounded representations of imperative sentences and propose a model for situated comprehension (in Section 5.2) that uses diverse knowledge to associate extra-linguistic context with imperative sentences and addresses several challenges. In Section 5.3, we describe the preliminary work on comprehending the imperative sentences and we conclude the chapter with a discussion on the future directions of the work. In Chapter 5 we describe how situated comprehension of instructions can be used to learn new action verbs in the domain via symbolic simulations.

5.1 Imperative Sentences

A typical imperative sentence is composed of a verb that indicates an action to be taken in the environment which is then instantiated with objects described with noun-phrases and relationships between objects using spatial prepositions. An example is - *Put the red large block in the pantry.* Successful comprehension of this linguistic construction involves identifying the referents of its linguistic components and combining them to generate an instantiated action representation. A referent is a physical or conceptual entity in the domain that is associated with linguistic items. The referents from our domain belong to three categories; objects ($\in R_o$, referents of noun phrases and pronouns), spatial relationships ($\in R_r$, referents of spatial prepositions) and action operators ($\in R_a$, referents of verbs). The action representation ($\in R_e$) is the referent of the sentence and is generated by composing the component referents.

The goal of comprehension of imperative sentences is to generate an instantiated action representation composed of the action's goal and its arguments and execute the corresponding policy.

An example is shown below

Instruction: *"Move the red, large, triangle to the left of the blue cylinder"*.
Interpretation: `ACT: operator OP1, argument1 O42, argument2 L42, goal.predicate REL42(O42,L42)`. (1)

where, the object `O42` satisfies the descriptive referring expression (RE) *the red, large, triangle*, the location `L42` satisfies the descriptive RE, *the pantry*, and the goal predicate `REL42` corresponds to the spatial composition associated with *to the left of* and is instantiated by the arguments of the action under constraints imposed by the verb *move* and its object REs. `OP1` is an abstraction over the execution policy corresponding to the verb *move*. This comprehension implements the mapping function -

$$M^{K}_{action}(i) : I_{AC} \rightarrow A$$

5.1.1 Ambiguities and Challenges

There are several challenges that have to be addressed in generating a grounded representation of an action. Several of them arise because natural language is ambiguous. Lexical knowledge and grammar of the language may be useful in rejecting several possible action representations that can be mapped to the language, however, typically such knowledge only under-constrains the representation. Evidence from cognitive linguistics (Piantadosi et al., 2012) suggests that ambiguity in language is useful in encoding a large amount of knowledge with fewer symbols. The extra-linguistic context these symbols are used in, provides the remaining constraints to generate an unambiguous representation. Both the speaker and hearer are aware of the extra-linguistic context of language and use it for generate meaning.

Other challenges arise because language may be used to convey information about objects and events that are not observable in the immediate present. The hearer has to rely on their past experiences in the environment and the general knowledge of the domain to ground such references. We describe the linguistic ambiguities and perceptual challenges that arise in learning with instruction below.

- **Linguistic Ambiguities**:

  CA1 Referring Expressions: The use of referring expressions to indicate the object of interest is context dependent. Consider the RE *red large triangle* and the object `O42` it refers to in example 5.1. Consider an environmental state in which the other object is smaller than `O42`. The instructor may refer to `O42` as the *large object* and omit information about color (*red*) or shape (*triangle*). Other contexts such as the context of the current task, attention, affordances of action may cause the instructor to use different REs including the severely under-constraining pronoun *it* to refer to the same object. Comprehension model in the agent should be robust to such context sensitive use of REs.

  CA2 Multiple Verb:Definitions: Several verb words map to different action definitions (and policies) in different context. For example, the imperative sentences *cook noodles* and *cook chicken* may map to different sequences of sub-actions despite of sharing the same linguistic structure. Linguistic information on its own does not provide enough constraints to determine the sequence of actions the agent needs to perform to successfully execute the command. Context derived from semantic organization in the domain and object affordances can inform sub-action execution.

  CA3 Verb, Preposition, & Action Argument Agreement: Consider the sentence in example 5.1. Successful execution of the action described by the sentence requires the agent to correctly instantiate the spatial relationship `REL42` (corresponding to *to the left of*) such that it is in agreement with the argument structure of the verb *move* (`REL42(O42,L42)` instead of `REL42(L42,O42)`). This constrain in not explicit in the linguistic structure of the sentence and has be derived from the domain models of the environment and prior experience with the verb.

  CA4 Preposition Phrase Attachment: The decision regarding the site of attaching the preposition phrase is ambiguous in English language. For example, the sentence, *move the red large object in the pantry to the right of the green cylinder*, is ambiguous. It could mean either *move (the red large object) (to the right of the green cylinder in the pantry)*, or *move (the red large object to the* *right of the green cylinder) (in the pantry).* Such ambiguities can be resolved by analyzing the current situation and reasoning about the likely interpretation.

- **Perceptual Challenges**

  CC1 Unobservable Referents: In domains with distance-limited sensing (like the real-world), several references to objects, locations, buildings etc. cannot be immediately grounded to current perceptions. However, successful action requires sensory information for the absent referents. For example, the to execute the command *go to the kitchen*, the agent needs to generate a locomotive plan and requires the agent to know where the *kitchen* is at. The agent must rely on prior experiences with the environment to provide such information. Similarly, several instructions might be relevant to states that slightly deviate from the currently observable state (efficient instruction; the instructor does not have to wait for the situation to arise but can provide instruction for hypothetical states). Interpretation of and learning from such instructions requires access to sensory information available from prior experiences with the environment.

  CC2 Past Events: In a highly dynamic, partially observable environment, the instructor is likely to commit observation or instruction errors. Instruction is useful in correcting behavior or adapting to new situations. In corrective instruction, the instructor observes agent's performance and provides better alternatives later. Comprehension of such instruction requires the agent to be able to resolve references to events that occurred in the past.

5.2 Grounded Comprehension Model for Imperative Sentences

Our comprehension model (Mohan and Laird, 2012c) is a formalization of the Indexical Hypothesis proposed by Glenberg and Robertson (1999) and generates a situation model for instruction using the available non-linguistic context and knowledge of the environment. The Indexical hypothesis explains how sentences become meaningful through grounding their interpretation in situated action. The hypothesis asserts that comprehending a sentence requires three processes: first, *indexing* words and phrases to referents that establishes the contents of the linguistic input, second, deriving *affordances* for these referents using knowledge of the environment, and third, *meshing* these affordances under the guidance of physical and task-specific constraints along with constraints provided by the syntax of the sentence (in Figure 2). The empirical evidence from human subjects in experimental studies (Kaschak and Glenberg, 2000) supports the Indexical Hypothesis. The linguistic information specifies a general scene, and the affordances of objects are used to specify the scene in detail sufficient to take action.

The following sections describe several aspects of the comprehension model including various kinds of situations that may arise from comprehension, sources that provide candidate referents, and diverse contexts that identify and constrain the possible referents.

5.2.1 Immediacy of Situation

Barsalou (1999) argued that language does not always describe (or ground to) situations that are immediate. In fact, most language comprehension tasks (including reading stories) rely on comprehenders to use referents from their long-term knowledge about the world and their prior experiences to create novel situations that are not accessible through immediate sensing. The ability of the comprehenders' to use their prior knowledge for resolving referents facilitates using shared experiences as teaching examples or creating hypothetical situations for learning. Linguistic instructions can index situations in the following ways.

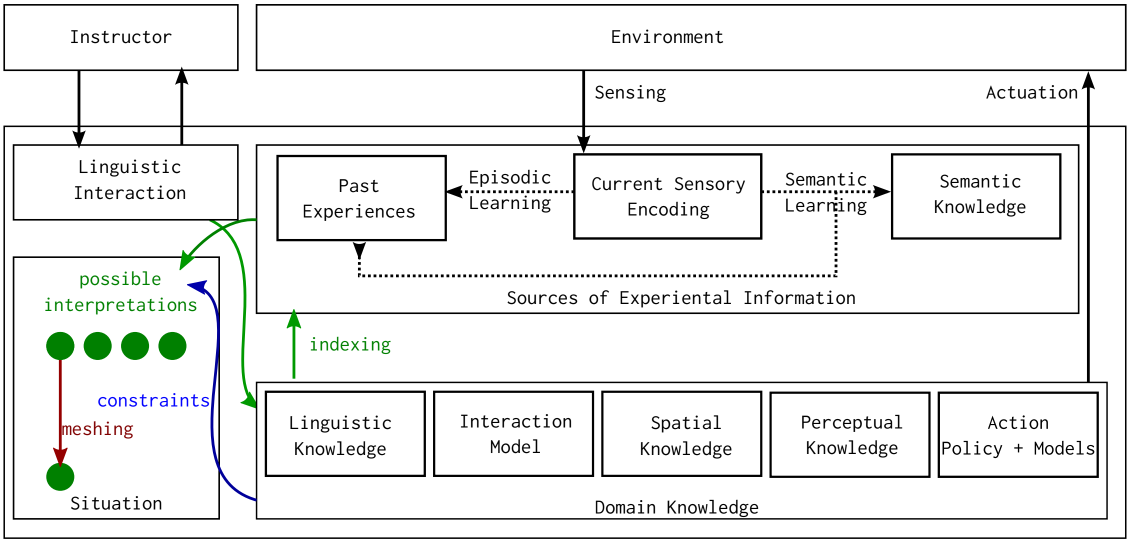


Figure 2: Comprehension Model for Imperative Instructions

- Immediate: Both the instructor and the learner agent are embedded in shared environment and can simultaneously view the physical situation under discussion. The instructor can refer to specific examples from the environment that are used by the agent to learn about the domain. (This has been implemented and has been shown to be useful in learning action verbs.)
- Displaced: In displaced indexing, the instructor and the agent discuss the situation they experienced earlier. Indexing can be displaced to the
  - Past, in which the instructor, having observed an action sequence taken by the agent provides corrections, facilitating recovery from instructional or observation errors.
  - Hypothetical Future: in which the instructor creates hypothetical scenarios alternative to the currently perceivable situation to provide examples of behavior for situation that might arise in future.

  (This constitutes the substantial part of proposed work.)

Comprehension of linguistic instructions results in a context relevant situation derived from information from various sources. The components of the situation are specified by the linguistic input, and their sensory and affordance-based details are filled in by domain knowledge.

5.2.2 Referent Sources

In the agent design, different kinds of situations can be achieved by indexing to referents provided by the following sources.

- Sensory Perceptions: The current sensory perceptions provide object referents (R_o) for immediate indexing.
- Episodic Memory: The agent accumulates its experience with the environment in its episodic memory. The objects (R_o) and events (R_e) and related sensory information in past episodes not only

facilitate indexing displacement in the past (challenge CC2) for corrective instruction but also provide sensory information for generating hypothetical scenarios (challenge CC1).

- Semantic Memory: Several kinds of semantic knowledge about the domain including spatial compositions (R_r) and declarative knowledge about actions (R_a) are encoded in semantic memory providing referents that are useful in grounding spatial prepositions and verbs. The semantic memory also encodes facts about the world which might be referred to by the instructor (challenge CC2).

5.2.3 Contexts

The agent has been encoded with diverse knowledge that is useful in perceiving and acting in the world and communicating with the instructor. This domain knowledge (K in the grounding functions G^K) serves as extra-linguistic context that is useful in identifying the set of candidate referents in the instruction and constraining the referent set based on current perceptions, spatial arrangement, the context of the current task and corresponding interaction, and the affordance of actions. Knoeferle and Crocker (2006) identify two dimensions of the interaction between the linguistic and situated context - *informational* and *temporal*. The informational dimension indicates that along with language, hearers use perceptual information and domain knowledge for comprehension. The temporal dimension indicates that cognitive attentional processes are closely associated with utterance generation and comprehension. These dimensions are encoded in our agents as follows:

- Informational
  - Syntactic: The lexical and syntactic analysis of instruction results in a parse tree corresponding to the imperative sentence, and a classification that inform further processing of instruction. The constraints induced by *part-of-speech* tags on linguistic symbols limit the category of referents they can be mapped to. For example, the verb word *"block"* maps to action definitions, eliminating possible object referents that can be described by the noun *"block"*.
  - Perceptual: The perceptual classification of objects along perceptual categories such as *color*, *size*, and *shape* along with the mapping of these classes to linguistic symbols facilitates building up sets of possible object referents in the imperative sentence.
  - Spatial: Spatial knowledge of prepositions is encoded as compositions of spatial primitives. This knowledge can be used to resolve object references using spatial primitives, *pick up the red triangle on the right of the green cylinder.*. The ability to recognize useful spatial relationships between the objects in the scene is useful in constraining the site of prepositional phrase attachments (ambiguity CA4).
  - Verb:Action Mapping: Mapping knowledge encodes how the syntactical structure of the verb maps to arguments of the corresponding action. It also constraints how the goal of the verb is instantiated with the objects of the verb forcing an agreement between instantiation of goal predicates, action arguments, and verb objects (ambiguity CA3). Semantic selectional restrictions (in Section 5.1.1) on the arguments of the verbs also provide extra constrains for determining the object referents of REs.
  - Action: This includes the *applicability* and *termination* conditions of an action and a policy corresponding to the action. The *applicability* conditions of the action provide affordance-based constraints on referent objects (ambiguity CA1). When provided with an instruction *"put it down"* when the agent is holding an object `O42`, the pronoun *it* can be resolved to `O42` without any other information from the instruction.

- Temporal
  - Interaction: The agent uses its interaction model (described in Appendix 7) to represent and increment the context and the current focus of the conversation between the instructor and the agent. The discourse context is informative about the references made using under-constrained referring expressions (ambiguity CA1) such as pronouns.
  - Attention: Objects that have been brought to attention, either through linguistic or extra-linguistic means, but are not in the focus of the ongoing communication are usually referred to by demonstrative pronouns or demonstrative noun phrases (Gundel et al., 1993). The extra-linguistic means may include unexpected behavior and pointing by the speaker. To resolve such REs, the agent must maintain the history of references to objects its perceptions. The agent relies on architectural activations for providing attentional contexts.

5.3 Current Status

Our current work on linguistic communication has focused on design and partial implementation of comprehension model (Mohan and Laird, 2012c) for interpretation and execution of instructions. This model implements indexical comprehension via immediate situation in which the instructor and the agent are simultaneously embedded in the environment. The agent uses its syntactic, perceptual, spatial, action, and interaction knowledge to identify and constrain the candidate referent set for noun-phrases, prepositions, and verbs. A further extension of the model (Mohan et al., 2013) can comprehend the complete range of referring expressions including partial and complete noun phrases and demonstrative and personal pronouns. Our interaction model for manages a task-oriented mixed-initiative communication (Mohan and Laird, 2012b,a; Mohan et al., 2012a) between the instructor and the agent and is described in Appendix A. We show that these models are useful in combining the experiential sensory and control information with linguistic information in the instructions and the structure of human-agent interaction to acquire grounded representations of verbs (Mohan et al., 2012a).

5.3.1 Analysis and Evaluation

Our preliminary analysis of the model confirms that the agent is able to use context driven from different knowledge sources for resolving ambiguities in comprehension of imperative sentences. A way to formally analyze such processes is to evaluate agent's performance in different scenarios that vary that vary along different dimensions including the objects on the scene and their arrangement, agent' knowledge, agent's experience, and the interaction context. For each of these scenarios, a small corpus of imperative sentences is generated. These sentences under-constrain the referents by varying degrees. The agent is evaluated on how many *information-gathering* questions it asks for correct comprehension.

A partial formal analysis of the comprehension model was conducted to evaluate its ability to correctly ground a wide range of referring expressions. We generated a corpus of 25 instructor utterances that addresses different capabilities of the agent. The instructions referred to objects using varying form of referring expressions including including 12 instances of personal pronouns (such as *it*), 4 instances of demonstrative pronouns (such as *this*), 3 instances of demonstrative phrases (such as *that cylinder*), and 14 varying length noun phrases with different descriptive words (such as *the red cylinder*). We evaluated various models of comprehension that exploit different dimensions of language-context interaction. The baseline model *p* uses the context derived from perceptual semantics only. To obtain other models, we incrementally added other kinds of contexts. Model *p+t* exploits the restrictions derived from task knowledge along with perceptual semantics. Model *p+t+a* exploits the temporal dimension by encoding the attentional state. Model *p+t+a+d* encodes both the attentional and dialog states.

Each of the comprehension models was evaluated using the instruction corpus on different scenarios of increasing perceptual ambiguity in the environment obtained by adding distractor objects. The *ambiguity 1* scenario only contained the intended referent objects on the scene. *Ambiguity 2* contained distractor objects that were perceptually distinct (different colors, shapes) from the intended referents. *Ambiguity 3* contained distractor objects that were of the same color as the intended referent but of different shapes. *Ambiguity 4* contained distractor objects that were perceptually similar to the intended referents and required the use of spatial references.

The results from these experiments are in Figure 3. The graph shows the number of *object identification* queries asked by the agent while using different comprehension models in scenarios with varying perceptual ambiguity. The design allows the agent to obtain more information through dialog if the RE is ambiguous. The agent reliably integrates information provided incrementally over several interactions for resolution. Consequently, all REs were resolved correctly in all models in all scenarios. In comparison, co-reference resolution in Stanford CoreNLP Lee et al. (2012) that exploits only the linguistic information, failed to correctly resolve 10 (28.6%) references. These results suggest that grounded contexts are essential for robust comprehension in an embodied agent that learns with situated instruction.

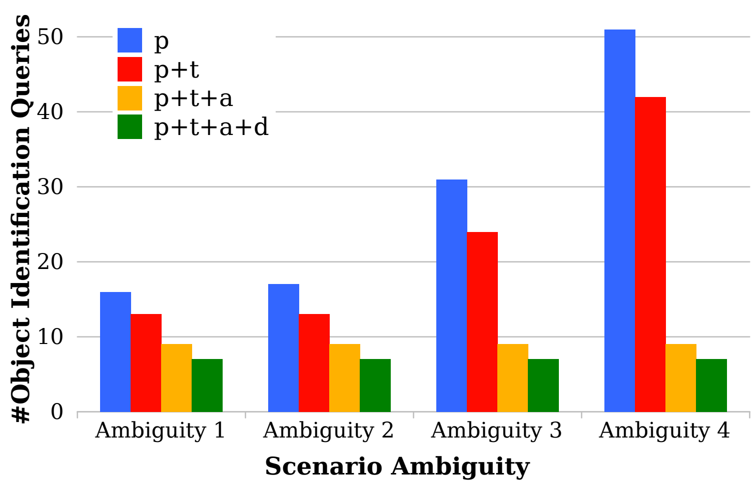


Figure 3: Number of Object Identification Queries

5.4 Future Work

Natural language can be used to indicate objects and events that are not perceivable currently but might have been a component of partly or completely shared experiences of the participants. This when combined with symbolic and perceptual simulations can be a powerful way to communicate information and share knowledge. This property of natural language allows the participants to learn from others' experiences. The speaker relies on the listener's past experiences with the environment and uses language to refer to those experiences. Such language can greatly enhance communication and instructions, since the participants are not limited to their limited immediate perceptions. This language can be comprehended through displaced situation of language or *displaced* indexing. We plan to extend the current comprehension model to support displaced indexing in the following conditions and tasks.

5.4.1 Partial Observability

Robot sensors have several limitations such as distance limited sensing that hinder the ability of the agent to perceive the complete world state. If the comprehension system is limited to resolving immediately perceptible referents, the agent will be unable to interpret action commands that refer to other referents that are observable or known to the instructor but not to the agent. This limits the communication in large domains such as the kitchen domain in Section 2.1. The agent is endowed with long-term declarative episodic and semantic memories which are built up linguistic and sensory experience in the environment. These memories are useful in virtual sensing that allow the agent to use its visual and sensory experience to reason about the past. These memories provide grounded context for resolving references to objects and events that are not currently perceptible but have been experienced in the past. Resolving referents to past knowledge of the environment expands the comprehension capabilities of the agent.

5.4.2 Hypothetical Situation

We are investigating ways that reduce the number of interactions required to learn, making learning with instruction efficient. One such way is to allow the instructor to provide instructions for situations that slightly deviate from the current state. The instructor does not have to wait for these situation to arise in the environment to provide instructions. The instructor can use natural language to create a hypothetical scenario that deviates from the current scenario, and instruct the agent in it. Interpretation of such instructions requires the agent to access to sensory information available from prior experiences with the environment and append them to current sensory perceptions to generate hypothetical scenarios.

5.4.3 Corrective Instruction

We currently assume that the instructor has perfect information about the environment state and is unlikely to make any instruction errors. However, this assumption does not hold for instructions in complex tasks in partially observable environments and for novice instructors. Several kinds of errors such as *observation errors* by the instructor or the agent, *instruction errors* by the instructor, or *interpretation errors* by the agent may arise. These errors may result in incorrect acquisition. A way to address these errors is to provide corrective instructions after observing behavior. Such corrective instruction exploits long-term memory in both participants. The instructor, while observing behavior, notes when agent's actions diverge from the expected behavior. The agent's episodic memory stores the sequences of actions it has taken in the environment. After the behavior is executed, the instructor and the agent use language to identify the incorrect parts of the behavior. The instructor then provides alternatives which are used by the agent to modify its behavior (further discussed in Section 6).

6 Learning Action Verbs from Instructions

Verbs encode complex syntactical, semantic, relational, and behavioral knowledge. Gentner (2006) argues that *"the noun class has the privilege of naming the highly cohesive bits of the world, whereas verbs and prepositions have the job of partitioning the leftovers-a diffuse set of largely relational components"*. In other words, many concrete nouns and adjectives refer to naturally bounded referents, perceptible in the world. In contrast, even very concrete verbs such as those that describe actions, stand for complex relationships between their objects, goals, and execution policy. Acquisition of verbs, therefore, is a significant challenge (and the second question that this thesis addresses), where the learner must induce hypotheses about how the objects of the verb are related and how these relationships affect behavior.

We intend to study *action verb* and action acquisition from the perspective of computational systems that are embodied in the real world and embedded in tasks-oriented interactions. Our long-term goal is to develop adaptive agents that can be taught to perform new tasks and procedures by their human collaborators, expanding their skill-set. Acquisition of verbs allow the agents to communicate about novel tasks. This is useful for future collaboration facilitated by linguistic communication and customization of their behavior. We assume that the agent begins with grammatical knowledge of English and with a simple lexicon that allows it parse imperative sentences. A substantial prior work in the natural language community and computational linguistics has looked at how the lexicon and the grammar can be acquired. We focus on acquisition of *grounded* representations of verbs that allow the agent to map an imperative sentence to an action representation.

The grounded representations of verbs must represent the meaning of the verb. They must be defined precisely, capturing both the predicate structure of the verb and the semantic content. These representations should be useful in identifying and constraining the action instantiations the command refers to. Through agent-prompted interactions with human, the agent also acquires task execution knowledge corresponding to the action verb. This includes the goal of the task and a high-level action policy over known primitive actions.

6.1 Verbs and Actions

The focus of this thesis is on acquisition of novel task knowledge in the environment along with learning how to communicate about such tasks linguistically in a human-agent interaction. This requires a commitment to verb representations that encode task goals and execution policies and are learned incrementally through interactions with human. Prior work on verb learning is largely silent on what knowledge should be encoded such that the linguistic verb structure is grounded in action representations and how such situated verb representations can be acquired. However, various works has addressed different parts of the problem, ranging from semantic encoding of verbs to learning goals and actions for novel tasks.In this section, we describe VerbNet in Section 6.1.1), a taxonomy of verbs that partially encodes the semantic information associated with verbs. We also present the different types of goals (in Section 6.1.2) that are useful for defining behavior in task-oriented agents. Finally, we describe prior work on acquiring task execution knowledge (in Section 6.1.3). Together, these set up requirements on what knowledge should be acquired to learn verbs that can be used to communicate about tasks. In later sections, we propose a representation scheme (6.2) and describe a preliminary acquisition scheme (6.3) that combine these ideas for comprehensive, assimilative learning and establish the future directions for this thesis.

6.1.1 VerbNet

VerbNet (Schuler, 2005) is a lexicon that organizes and categorize various English verbs on the basis of their common syntactic and semantic properties. It has been used for several linguistic tasks including dialog, verb sense dismabiguation, and concept network creation. The representation of verbs in VerbNet captures both the syntax and semantics of the verbs, and makes explicit links between the two. This knowledge can be useful in associating verbs with semantics of actions. VerbNet is a class based lexicon in which each class is defined by, VN1 Thematic Roles: Thematic roles (23 in VerbNet) refer to the underlying semantic relationship between a verb's predicate and its arguments. These roles are used to describe lexical and semantic patterns in the behavior corresponding to the verb. Roles relevant to this thesis include

- `patient`: a participant whose state is changed as an effect of the action.
- `theme`: a participant that is located at a place and is seen moving from one place to the other.
- `instrument`: a physical force or an object that causes a change in state of another object.
- `location, destination, source`: spatial locations of objects.

VN2 Selectional Restrictions: Selectional restrictions constraint which objects can be used as arguments for verb predicates. Although, VerbNet only encodes semantic selectional restrictions on objects of verbs, such restrictions may arise from other sources of knowledge (Kamide et al., 2003). In our agent, these sources may include affordances, semantic knowledge, domain knowledge, current sensory perceptions etc.

VN3 Syntactic Frames: Syntactic frames (in Example 6.1.1) are surface realization for a verb. They describe constructions such as transitive, intransitive, prepositional phrases, etc. A syntactic frame consists of the thematic role in their argument position around the verb, the verb, and other supporting lexical items. Additional restrictions may be included based on number agreement and syntactical restrictions.

Agent v Patient
(Bob hits the ball.) (2)

VN4 Semantic Predicates: Semantic predicates encode relational information about the objects of the verb including their state, spatial configuration, movement, manner, and time.

cause(Agent,E)
manner(during(E), directedMotion, Agent) (3)
...

VerbNet provides a domain-general representation of English verbs. Although, it represents information about actions corresponding to verbs through semantic predicates, these are not completely defined. It also does not encode the behavior policy corresponding to verbs, or how verbs are connected to spatial and perceptual information through prepositions and nouns/adjectives. However, the representations in VerbNet are suggestive of what knowledge should the agent acquire to learn linguistic and semantic aspects of verbs.

6.1.2 Action Goals

The agent is embedded in a task-oriented environment and learns novel task execution knowledge, therefore, the acquistion and representation of task goals play an central role in agent design. Braubach et al. (2005) discuss various types of goals that are useful in designing agents that demonstrate intelligent behavior. The following dimensions have been adapted from their work and will be used to explore different types of goals in this thesis.

AG1 Goal predicates: Several types of object attribute contribute to defining the goal of an action. In this thesis, we will focus on the following.

- *Spatial* predicates define the spatial configuration of objects in the work space. Examples of spatial predicates include, *right of*, *in front of*, *between*, etc.
- *Know-value* predicates motivate information gathering actions for properties that cannot be visually observed in the environment, but require dedicate actions that may involve using instruments. An example information gathering action is *weigh* that establishes the weight of an object. This action is executed if the agent does not know the weight of the object.
- *State-change* predicates observe and track the changes in the state of objects (*patient* in VerbNet representation) as a result of actions. Verbs such as *cook* have goals that monitor goals based on state changes of objects that are being *cooked*.

AG2 Composition: To define a goal, the attributes can be composed together using conjunction, disjunctions, or negations.

AG3 Temporal behavior: This dimension pertains to the relationship between actions and their goals. Goal types relevant to this thesis include, *achievement goals* that require the agent to take action to achieve the set of goal predicates, *maintenance goals* that require the agent to observe and maintain some world state defined by the goal, and *perform* goals pertain to an activity instead of meeting the goal conditions.

These goals inform what semantic predicates should be encoded while representing verbs in agents that are embedded in a task-oriented environment. While semantic predicate representations in VerbNet is more comprehensive and encodes temporal and causal information, we will focus on the semantic predicates that allow the agent to represent various kinds of goals associated with action verbs useful in the domain. We rely on the procedural knowledge of an action to provide causal constraints and information for verbs.

6.1.3 Task Execution

Huffman and Laird (1995) have previously explored how new task-oriented procedural knowledge can be acquired through instruction. Their work focused on exploring different strategies in situated EBL framework that can be used to acquire generalized knowledge from specific examples and their implications on learning time. They identified the key knowledge categories the agent must acquire in order to learn new tasks.

TE1 State inference: These rules implement simple monotonic inferences. Such rules augment the representation of the agent's state by inferring the state properties based on the current state descriptors. For instance, the agent know that it is not holding anything if the gripper is empty.

TE2 Operator selection: The agent must select an operator to apply given the current state of the environment. This process involves two types of knowledge.

- *Proposal knowledge*: These rules encode the availability conditions/pre-conditions of an action.
- *Control knowledge*: These rules assign a preference order to all the operator available in the current state.

TE3 Operator application: A selected operator is applied to the current state either by directly applying the actions of a rule to the current state (motor commands) or by invoking a policy over sub-operator selection (above).

TE4 Operator termination: An operator/action must be terminated when its application is complete. The termination of the operator is dependent on the goal of the action (Section 6.1.2).

6.2 Action-oriented Verb Representation

Although, VerbNet represents structured information about various aspects of verbs, it does not provide enough details for the agent to take action in the environment. We now propose representations that encode the knowledge represented in VerbNet along with domain specific details that allows the to act in the world. This *action-oriented* representation of verbs encodes goal-oriented behaviors along with syntactical and semantic information of verbs and creates associative links between verbs, prepositions, and objects (requirement R3b). Through a general acquisition process, these representations can be acquired through interactive instruction (Mohan et al., 2012a,b). The representation allows the agent to encode various kinds of verbs that differ in syntactical and linguistic structure, semantic knowledge, and policy (requirement R3c).

The representation of verbs is distributed across three knowledge categories (requirement R3a): linguistic *mapping* knowledge that encodes syntactical information and how the surface realization of verbs maps to actions, *semantic* knowledge that encodes how the arguments of the verb are related to the action goals, and the *procedural knowledge* that encodes behavior policy grounded in agent's sensory perceptions and control mechanisms. The details of these representations and their correspondence to VerbNet components is discussed below.

6.2.1 Map

A linguistic *map* (in Figure 4) associates a verb and its argument structure to an action and objects in the environment. This encodes a subset of knowledge K that is useful in the grounding function G^K_{action}, for imperative sentences. The map is encoded declaratively in the agent's semantic memory as a semantic network. The maps encode the surface syntactic realization of verbs and serve a function similar to syntactic frames of VerbNet. Thematic roles are not explicitly encoded in this representation but are implicitly represented in the task implementation knowledge (in Section 5.2.3).

The linguistic maps are useful situated comprehension of verbs. It provides the verb-action argument agreement, constraining the possible interpretations. To use maps, the agent queries its semantic memory using a cue created from the verb and its argument structure in the imperative sentence it is attempting to comprehend. The semantic memory returns the related action operator which can be instantiated with objects in the environment under the constraints of the map. Consider the example shown in Figure 4. It maps the verb *put* with an argument structure consisting of a *direct-object* (Node AO1) and the object connected to the verb via the preposition *in* (Node A02) with the behavior operator `op-put-down`. The nodes A01 and A02 are slots that can be filled up the objects in the environment that satisfy the REs. This allows the agent to associate the sentence *Put a red, large block in the dishwasher* with an appropriate operator and instantiate it with objects such that it will achieve the intended goal.

The agent begins with map representations for known primitive verbs. It can acquire maps for the novel verbs. The novel maps are created using the information from the syntactical parser when the agent acquires a new verb and action. One such map is shown in Figure 5 (nodes (`M1`, `L1`, `A11`, `P1`)). This mapping

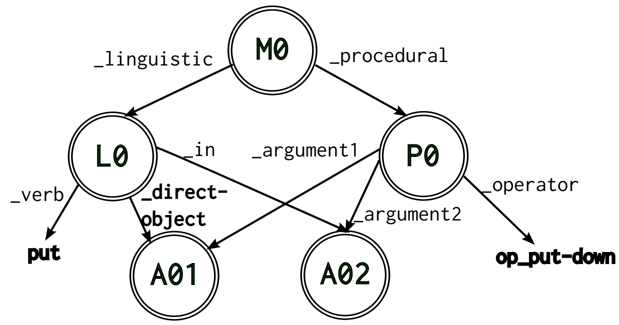


Figure 4: Linguistic Map for the verb *put*

associates the novel verb *store* and its *direct-object* to a new operator (`op_1`) and its argument. `A11` is a slot that can be filled by an object that satisfies the description (noun-phrase) connected to the verb as a *direct-object*. corresponds to *direct-object* of the verb. The edge `A11,P1` constrains the instantiation of `argument1` to the same object. Future action commands containing the verb *store* can be indexed to the action `op_1` using this map. Note that if a verb is used refer to multiple action representations, the agent will acquire multiple mapping structures. This will cause a CA2 ambiguity during comprehension.

6.2.2 Action Concept Networks

The action concept network (ACN, in Figure 4) encodes the declarative semantic information about the verb and the corresponding action. The ACN creates constraints between the linguistic knowledge of the verb and the compositional structure of the preposition *in* and the location *pantry*. These concepts may have been acquired in previous experiences or may be learned with the verb acquisition. Thereby, the ACN serves to integrate the linguistic and semantic knowledge of verbs with prepositions and objects, thus assimilating diverse kinds of knowledge (requirement R3b), acquired at different points in the agent's life time.

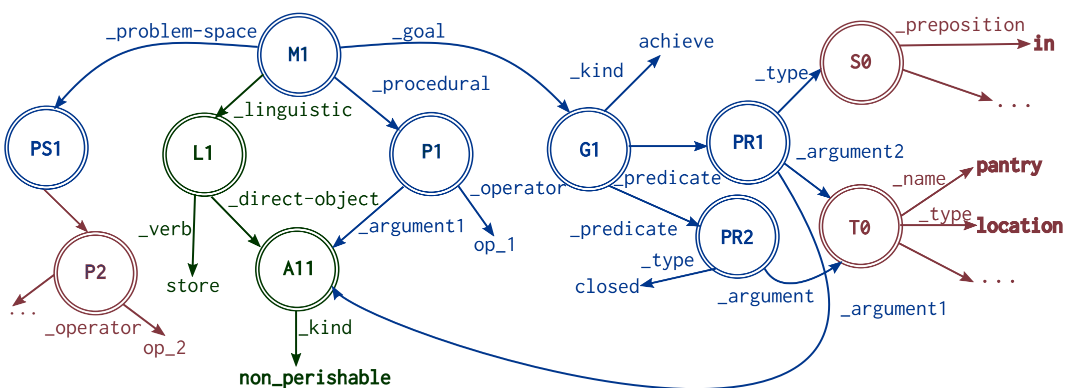


Figure 5: Linguistic Map and Action Concept Network for the verb *store*

The ACN encodes the following knowledge:

- **Goal concept**: The goal concept is declaratively represented in semantic memory as a part of the action concept network. It is a composition of goal predicates that may including spatial compositions denoted by prepositions known to the agent (shown in Figure 4). This concept is useful in identifying the termination conditions P^a_{term} of an action a.

- **Implicit parameters**: Some verb commands such as *move the red block to the pantry* make the argument structure *explicit* by including the object and the location in the command. However, in other verb commands such as *store the red block*, an argument (`location` = `pantry`) might be *implicit* in the verb *store* itself. Implicit parameters are learned through instruction and `memorized' by storing them in the semantic memory with appropriate connections to the verb and the action.

- **Verb, preposition, action argument agreement**: Several nodes in the ACN such as `A11` represent slots that can be filled up the objects referred to by REs. Certain edges in the network define constraints on how the actions and goals may be instantiated. The edge `P2,A11` introduces constraints on how the goal can be defined on the basis of instantiation of the verb. The edge asserts that `_argument1` of `_goal_predicate` of type `in` should be instantiated with the `_direct-object` of the verb (`_argument1` of the action). These constrains not only correctly define the goal (given an *imperative sentence*), they are useful in reducing the CA3 ambiguity during comprehension.

- **Semantic selectional constraints**: The ACN can also encode semantic selection constraints over which objects can be used to instantiate an action (corresponding to the verb). Consider the slot `A11`, which can be filled by an object that satisfies the description in the referring expression (in the *direct-object* of the verb) and is also a `non_perishable` object. The network defines the action of storing an object in the pantry only for objects that are `non-perishable`. These semantic constraints are also useful in resolving CA1 ambiguities during comprehension. When the verb arguments are under constrained REs, the agent can use its domain semantic knowledge to constrain the set, possible referent objects.

- **Action space**: Sub-graph (M1, PS1, P2, ...) specifies the action-space (A^a_c) of the verb *store*. In the example, this set contains the action `op_2` corresponding to the verb *move* (not shown in figure) that has been acquired by the agent from previous experience with the instructor. Further connections between the ACNs of *move* and *store* constrain how the behavior policy and the goal of *move* is instantiated with respect to the arguments of *store*. The knowledge of the problem space of an action is useful in inducing the *availability* conditions of the consituent actions.

6.2.3 Behavior

Behavior or task execution knowledge is encoded in our agent as rules. Behavior acquisition has been adapted from (Huffman and Laird, 1995). We discuss the various categories of knowledge that our agent acquires below and describe how they are useful in comprehension and execution of verbs.

- **State inference**: These rules only implicitly assist in execution and acquisition of action verbs. We will pre-program these rules and will not study their acquisition.

- **Availability conditions**: An important part of knowing an action (verb) is knowing when that action is available to be executed in the environment. In our agent, the availability conditions (P^a_{avail}) are encoded as operator proposal rules. These rules propose action operators which may later be selected for execution. Availability conditions (operator proposal rules) incorporate knowledge of object affordances. The proposal rules of the `pick-up` operator test if there is a movable object in the environment. These rules convey important information about the corresponding verb. They implement *affordance-based* selectional restrictions on the action arguments. This is useful in comprehension of imperative sentences when the referring expression is under constrained. The agent can reason about which interpretation of the sentences can be applied to the current situation given the actions it can take. This thesis will investigate different interactions through which these rules can be acquired.

- **Application rules**: These rules apply the action to the current state. These rules may affect the state by either executing the motor commands in the environment, or through a deliberate step in internal reasoning. They encode the following information about the verbs.
  - *Thematic roles*: The application rules determine how the objects in the environment will be manipulated, implementing the thematic roles implicitly. For example, in response to *move the red triangle to the pantry*, the `move` operator causes the robot to use its gripper to pick the *red triangle* up and place it at a different co-ordinate in the *pantry*. In this case, the object *red triangle* serves the role of *theme* of the verb *move*.
  - *Semantic predicates*: The application rules also test for the presence or absence of certain relationships, and change the policy based on these relationships. For example, to *clear the pantry*, the robot might move all the objects such that no object satisfies `in(<obj>, pantry)` semantic predicate.
- **Policy**: These rules encode action selection policy (π^a) - which actions should be selected for executed given a set of available actions in the state. This selection knowledge can be represented in Soar either has symbolic preferences or numeric preferences that can be updated with reinforcement learning.
- **Termination conditions**: Termination conditions (TE4) determine when the action has been successfully applied. They are influenced by the what kind of goal the agent is pursuing. For *achievement* goals, the termination conditions of an action are when the goal conditions are met. This requires the agent to continuously track goals as it is executing the action. For *perform* goals, the agent generates a plan to achieve the goal state and executes the plan without tracking if the goal conditions are met. Actions involving *maintenance* goals terminate when the goals is re-established.

6.3 Acquisition

Verb and action acquisition occurs during performance of the agent in the task and is integrated within linguistic interactions with the human collaborator (integrative interaction R2a). The process begins with *lexical processing* of the human' action command. The agent, then, increments the interaction state and attempts to generate a situated representation of the action command. On successful interpretation, the command is executed in the environment. If the agent fails to generate a grounded interpretation or to execute the action in the command, it begins dedicated interactions (*active* learning R3e) with the human instructor to learn the missing knowledge. Acquisition of knowledge in all categories is driven by *knowledge failure* i.e. failure of the agent to progress due to lack of knowledge. The failures include *impasses* and long-term memory *retrieval* failures. Although, we only discuss acquisition of verbs in this thesis, the agent can acquire perceptual, semantic, and linguistic maps for nouns and adjectives, and spatial compositions and linguistic maps for prepositions using this interaction and learning scheme (Mohan et al., 2012a).

- **Maps**: The *maps* are acquired through interactions initiated by the agent when it fails to access an action concept network corresponding to the verb used in the imperative sentence. This failure indicates that either the imperative sentence contains a new verb, or a novel argument structure is used with a known verb. In both cases, the agent generates a new map. The linguistic sub-network (Figure 5 sub-graph (M1, L1, A11)) of the map is generated using the information available from the linguistic parse of the sentence. The procedural sub-network (Figure 5 sub-graph (M1, P1, A11)) is generated by the agent and is associated with a new operator. This map is stored in agent's semantic memory. Note that even though the map is generated from a specific imperative sentence, it encodes general knowledge (*general* R3c) by creating slots for the arguments of the verb.

- **Action Concept Network**: Acquisition of *maps* allows the agent to associate the verb to the action operator. However, since the agent lacks the semantic knowledge of goal predicates, semantic memory retrieval fails. The agent attempts to acquire these missing pieces through dedicated interactions. First, the agent poses a dedicated query about the goal of the verb. After the reply from the instructor is successfully grounded, the agent assimilates the reply to create the ACN (Figure 5 sub-graph (P1,G2,P2,S0,T0)). In the ACN, the agent creates constraints between the verb arguments, action operator instantiation, goal predicates and spatial relationships. This not only serves to create explicit associations between the knowledge acquired at different stages in agent's life (*assimilative* R3b) but also generalizes the concept of the goal (*general* R3c).

- **Behavior**: Even after acquiring the goal concept, the agent cannot perform the task since it lacks a corresponding policy. The current design of the agent acquires this policy by obtaining an example policy execution through interaction. The agent prompts the instructor for the next sub-action it should take to move closer to the goal of the task. The sub-action is added to the problem-space of the novel verb. Through multiple such interactions, the agent is able to achieve the goal in the environment. The agent then attempts to extract a general execution policy from this example behavior through situated explanation based on complete retrospective instruction recall. This results in new rules corresponding that implement partial/complete policy for this novel behavior along with applicability conditions of the constituent actions. Even with a few examples, the agent is able to acquire selection rules that implement a major portion of the policy (*fast* generalization R3c). This learning scheme allows the agent to exploits the knowledge acquired through interaction immediately towards task performance.

6.3.1 Preliminary Analysis

To evaluate the performance of our agent, we generated a space of action command templates from four novel verbs – three each for the verbs *move* and *shift* by combining them with prepositions *to*, *to the left of* and *to the right of*, and one each from *store*, and *discard*. For training, these templates were instantiated with an object and a location to generate commands which were given to the agent. If the agent asked any questions, instructions were provided from a set of relevant goal descriptions and action commands. For testing, we generated two commands by instantiating a template with randomly selected objects (four in the scene) and locations (four in the scene). These commands were tested in different initial world states (`hold[obj]/empty` & `open[loc]/close[loc]`).

Partial results summarized in Table 1 show the different kinds of verbs (and corresponding actions) the agent can acquire through the mechanisms described here. The commands have different kinds of arguments (explicit v/s implicit), different goal predicates (spatial/state), and different policies. Column I records the number of possible instantiations of a template, S records the number of possible initial world states, and T records the number of interactions in the single training sequence for that verb and action policy. The knowledge acquired by the agent has the following properties:

- **Generality of arguments**: Since the agent acquires general action-concept networks and behavior policy templates, the knowledge learned during the single training instance generalizes to other objects and locations in the scene. This was tested by giving randomly instantiated commands for learned verbs to the agent and verifying that they were correctly executed.

- **Generality of instruction sequence**: During explanation, the agent determines why a sequence of actions was useful for achieving the desired goal of the verb. Superfluous actions in the training sequence, those that were not useful in progress towards the goal, are not included in the execution policy. For example, if the training sequence of *move the blue cylinder to the garbage* was *pick up the blue cylinder*, *put the cylinder in the pantry*, *pick up the cylinder*, *put the cylinder in the garbage*, the agent's policy for `move[obj]`, will be `pick-up[obj]`, `put-down[in, obj, loc]`.

- **State sensitive execution**: From a single training sequence, the agent acquires a policy that applies to any initial state of the world. If the agent is executing store, and the pantry is open, it does not open the pantry again, but moves the relevant object to the pantry. This was tested by giving the command in all initial states and verifying that the agent successfully executed the command.

- **Policy transfer**: The agent can acquire hierarchical execution policies. This allows the instructor to aid in transferring policy from one verb to the others. For *shift*, *store* and *discard*, the instructor can either give a sequence of primitive actions (policy 1) or use an acquired composite verb *move* (policy 2) in place of *pick-up* and *put-down*. The hierarchical policy is taught in fewer interactions than the flat policy (column T: 11 vs. 13).

Table 1: Evaluation of Generalization

Template	I	Arguments	Goal	Policy	S	T
move [obj] to [loc]	16	both explicit	in(obj, loc)	pick-up(obj), put-down(in, obj, loc)	2	9
move [obj] to the left of [loc]	16	both explicit	in(obj, loc)	pick-up(obj), put-down(left, obj, loc)	2	9
move [obj] to the right of [loc]	16	both explicit	in(obj, loc)	pick-up(obj), put-down(right, obj, loc)	2	9
shift [obj] to [loc]	16	both explicit	in(obj, loc), closed (loc)	1. open(loc), pick-up(obj), put-down(in, obj, loc), close(loc)	4	13
				2. open(loc), move(in, obj, loc), close(loc)		11
store [obj]	4	[loc] implicit	in(obj, pantry), closed (pantry)	open(pantry),move(in, obj, pantry), close(pantry)	4	11
discard [obj]	4	[loc] implicit	in(obj, garbage)	move(obj, in, garbage)	4	7

6.4 Summary

The following table summarizes the kinds of knowledge that can be encoded using the representation introduced in Section 5.2. For comparison, we also show what can be encoded in VerbNet and InstructoSoar by Huffman and Laird (1995). ✓ stands for if a particular category of knowledge can be encoded in the representation, × stands for non supported knowledge. ∗ represents if the work investigates acquisition of the knowledge category.

Table 2: Comparative Summary of Knowledge Representation in Prior Work

Category	Type	VerbNet	InstructoSoar	This thesis
Linguistic	syntactical	✓	×	✓ ∗
	grounding	×	×	✓ ∗ (maps)
	selectional restrictions	✓ (semantic)	×	✓ ∗ (affordance) (semantic)
	thematic roles	✓ (explicit)		✓ ∗
Semantic	predicates	✓ (semantic)	✓ ∗ (procedural)	✓ ∗ (procedural) (semantic)
	goal kind	×	×	✓ ∗ (semantic)
	goal composition	×	×	✓ ∗ (semantic)
Procedural			(procedural)	(procedural)
	state inference	×	✓ ∗	✓
	availability	×	✓ ∗	✓ ∗
	termination	×	✓	✓ ∗
	execution	×	✓ ∗	✓
	control policy	×	✓ ∗	✓ ∗

6.5 Future Directions

Our preliminary work has primarily focused on analyzing the requirements of an instructable agent (Mohan and Laird, 2011), identifying the representations necessary for encoding the diverse knowledge agent acquires and implementing and analyzing the basic mechanisms required for learning with interactive instruction (Mohan et al., 2012a,b). Future work will be carried along the following dimensions.

6.5.1 Structural-level Extensions

We plan to implement extensions to the current work such that the agent is able to acquire knowledge in categories identified in the previous sections (marked in Table 3). These include

- Goal concepts: Current implementation supports acquisition of *achievement* goals which are compositions of spatial predicates. Future implementations will expand this to acquisition of all types of goals in the taxonomy in Section 6.1.2 and corresponding termination conditions.
- Semantic selectional restrictions: Several actions (and corresponding verbs) have semantic selection restrictions. Such as, only non-perishable objects can be *stored in the pantry*. The constraints pro-

vided by these restrictions not only constrain the candidate object set for verb arguments, they also inform which action representation should be selected for execution. For example, when the verb *store* is used for a perishable object, the agent might act to keep the object in the refrigerator. These restrictions can be learned by explicit interactions with the human instructor.

6.5.2 Learning from Hypothetical States

Humans are able to learn from language that is not grounded in the current sensory stream. Examples of such learning includes learning from instruction manuals or through hypothetical situations. The ability to learn from instructions that are not situated in the present sensory state allows the human learners to acquire useful knowledge *before* that situation arises in the environment. This is useful from the instructor's perspective too. This allows the instructor to provide information to the learner when it is convenient instead of closely monitoring the agent behavior and waiting for the situations to arise in the environment. Interpretation of and learning from such instructions requires the agent to access to sensory information available from prior experiences with the environment and create hypothetical situations to learn from. Perceptual simulations in these hypothetical situations using domain models can inform agent's learning.

6.5.3 Robust Learning

We currently assume that the instructor has perfect information about the environment state and is unlikely to make any instruction errors. However, this assumption does not hold for instructions in complex tasks. Noise may arise due to several factors including observation errors by both the agent and the instructor and communication errors by the instructor. We are interested in exploring learning from corrective instruction. This learning can be used to relearn the goal and consequently the policy, in case incorrect goals and policy were acquired. Corrections can also be used to tailor agent's behavior to the human collaborator's preferences by providing examples for a different policy to the same goal.

7 Timeline

Table 3: Timeline for completing the thesis work

Date	Milestone
May 2013	Implement structural-level extensions for learning (section 6.5.1)
June 2013	Extend system to work on distance-limited sensing domain (Section 5.4.1)
October 2013	Implement learning in hypothetical-situation (section 5.4.2 & section 6.5.2)
February 2014	Implement learning from corrective instruction (section 5.4.3 & section 6.5.3)
April 2014	Run evaluations and data collection
June 2014	Thesis writing

References

Allen, J., Chambers, N., Ferguson, G., and Galescu, L. (2007). Plow: A Collaborative Task Learning Agent. In *Proceedings of the National Conference on Artificial Intelligence*, Vancouver, BC.

Argall, B. D., Chernova, S., Veloso, M., and Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469--483.

Barnard, K., Forsyth, D., and Jordan, M. I. (2003). Matching Words and Pictures. *Journal of Machine Learning Research*, 3:1107--1135.

Barsalou, L. W. (1999). Language Comprehension: Archival Memory or Preparation for Situated Action? *Discourse Processes*, 28(1):61--80.

Braubach, L., Pokahr, A., Moldt, D., and Lamersdorf, W. (2005). Goal Representation for BDI Agent Systems. *Programming Multi-Agent Systems*, pages 44--65.

Byron, D. and Fosler-Lussier, E. (2005). The OSU Quake 2004 corpus of two-party situated problem-solving dialogs. Technical report, The Ohio State University.

Cantrell, R., Schermerhorn, P., and Scheutz, M. (2011). Learning Actions from Human-Robot Dialogues. In *Proceedings of the IEEE Symposium on Robot and Human Interactive Communication*, pages 125--130.

Cantrell, R., Scheutz, M., Schermerhorn, P., and Wu, X. (2010). Robust Spoken Instruction Understanding for HRI. In *Proceedings of the 5th ACM/IEEE International Conference on Human-Robot Interaction*, pages 275--282.

Chen, D. L. and Mooney, R. J. (2011). Learning to Interpret Natural Language Navigation Instructions from Observations. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, pages 859--865.

Chen, X., Ji, J., Jiang, J., and Jin, G. (2010). Developing high-level cognitive functions for service robots. In *Proceedings of 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pages 989--996.

Chernova, S. and Veloso, M. (2009). Interactive policy learning through confidence-based autonomy. *Journal of Artificial Intelligence Research*, 34(1):1--25.

Dinerstein, J., Egbert, P. K., and Ventura, D. (2007). Learning Policies for Embodied Virtual Agents Through Demonstration. In *In Proceedings of International Joint Conference on Artificial Intelligence*.

Gentner, D. (2006). Why verbs are hard to learn. *Action meets word: How children learn verbs*, pages 544----564.

Glenberg, A. M. and Robertson, D. A. (1999). Indexical Understanding of Instructions. *Discourse Processes*, 28(1):1--26.

Goertzel, B., Pennachin, C., Araujo, S., Silva, F., Queiroz, M., Lian, R., Silva, W., Ross, M., Vepstas, L., and Senna, A. (2010). A General Intelligence Oriented Architecture for Embodied Natural Language Processing. In *Proceedings of the 3d Conference on Artificial General Intelligence*.

Goertzel, B., Pennachin, C., Geissweiller, N., Looks, M., Senna, A., Silva, W., Heljakka, A., and Lopes, C. (2008). An Integrative Methodology for Teaching Embodied Non-Linguistic Agents , Applied to Virtual Animals in Second Life. In *Artificial General Intelligence*.

Gorniak, P. and Roy, D. (2004). Grounded Semantic Composition for Visual Scenes. *Journal of Artificial Intelligence Research*, 21:429--470.

Grosz, B. and Sidner, C. (1986). Attention, Intentions, and the Structure of Discourse. *Computational linguistics*, 12(3):175--204.

Gundel, J. K., Hedberg, N., Zacharski, R., and Fraser, S. (1993). Cognitive Status and the form of Referring Expressions in Discourse. *Language*, 69(2):274----307.

Gupta, A. and Davis, L. S. (2008). Beyond Nouns: Exploiting Prepositions and Comparative Adjectives for Learning Visual Classifiers. In *Proceedings of the 10th European Conference on Computer Vision*.

Huffman, S. and Laird, J. E. (1995). Flexibly Instructable Agents. *Journal of Artificial Intelligence Research*, 3.

Kamide, Y., Scheepers, C., and Altmann, G. T. M. (2003). Integration of Syntactic and Semantic Information in Predictive Processing: Cross-Linguistic Evidence from German and English. *Journal of psycholinguistic research*, 32(1):37--55.

Kaschak, M. P. and Glenberg, A. M. (2000). Constructing Meaning: The Role of Affordances and Grammatical Constructions in Sentence Comprehension. *Journal of Memory and Language*, 43(3):508--529.

Knoeferle, P. and Crocker, M. W. (2006). The Coordinated Interplay of Scene, Utterance, and World Knowledge: Evidence from Eye Tracking. *Cognitive Science*, 30(3):481--529.

Knox, W. B. and Stone, P. (2010). Combining Manual Feedback with Subsequent MDP Reward Signals for Reinforcement Learning. In *In Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems*.

Kollar, T., Tellex, S., Roy, D., and Roy, N. (2010). Toward Understanding Natural Language Directions. In *Proceeding of the 5th ACM/IEEE international conference on Human-robot interaction - HRI '10*, page 259, New York, New York, USA.

Laird, J. E. (2012). *The Soar Cognitive Architecture*. MIT Press.

Laird, J. E., Yager, E. S., Hucka, M., and Tuck, C. M. (1991). Robo-Soar: An Integration of External Interaction, Planning, and Learning using Soar. *Robotics and Autonomous Systems*, 8(1-2):113--129.

Lee, H., Chang, A., Peirsman, Y., Chambers, N., Surdeanu, M., and Jurafsky, D. (2012). Deterministic Coreference Resolution Based on Entity-Centric, Precision-Ranked Rules. *Computational Linguistics*.

Maclin, R. and Shavlik, J. W. (1996). Creating advice-taking reinforcement learners. *Machine Learning*, 22(1-3):251--281.

MacMahon, M., Stankiewicz, B., and Kuipers, B. (2006). Walk the talk: Connecting language, knowledge, and action in route instructions. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI '06)*, Boston, MA.

Matuszek, C., FitzGerald, N., and Zettlemoyer, L. (2012). A Joint Model of Language and Perception for Grounded Attribute Learning. *Internation Conference on Machine Learning*.

Mohan, S., Kirk, J. R., Mininger, A., and Laird, J. (2012a). Acquiring Grounded Representations of Words with Situated Interactive Instruction. *Advances in Cognitive Systems*.

Mohan, S. and Laird, J. (2012a). Situated Comprehension of Imperative Sentences in Embodied, Cognitive Agents. *Workshops at the Twenty-Sixth AAAI Conference on ...*, pages 25--31.

Mohan, S. and Laird, J. E. (2011). Towards Situated, Interactive, Instructable Agents in a Cognitive Architecture. In *2011 AAAI Fall Symposium Series*.

Mohan, S. and Laird, J. E. (2012b). Exploring Mixed Initiative Interaction for Learning with Instruction in Cognitive Agents. In *Twenty-Sixth Conference on Artificial Intelligence (AAAI Student Abstract)*.

Mohan, S. and Laird, J. J. (2012c). Situated Comprehension of Imperative Sentences in Embodied, Cognitive Agents. In *Technical Reports from AAAI Workshop on Grounded Language for Physical Systems*, pages 25--31.

Mohan, S., Mininger, A., Kirk, J., and Laird, J. (2012b). Learning Grounded Language through Situated Interactive Instruction. In *Technical Reports from AAAI Fall Symposium Series*.

Mohan, S., Mininger, A., and Laird, J. (2013). Referring Expression Resolution using Diverse Contexts for Instructable Embodied Cognitive Agents. In *Submissions to the Annual Conference of the Association of Computational Linguistics*.

Oviatt, S. L. and Cohen, P. R. (1991). Discourse structure and performance efficiency in interactive aud non-interactive spoken modollities. *Computer Speech & Language1*, 5(4):297--326.

Piantadosi, S. T., Tily, H., and Gibson, E. (2012). The communicative function of ambiguity in language. *Cognition*, 122(3):280--91.

Rich, C. and Sidner, C. (1998). COLLAGEN: A Collaboration Manager for Software Interface Agents. *User Modeling and User-Adapted Interaction*, 8(3):315--350.

Roy, D. (2002). Learning Visually Grounded Words and Syntax for a Scene Description Task. *Computer Speech & Language*.

Sammut, C., Hurst, S., Kedzier, D., and Michie, D. (1992). Learning to Fly. In *Proceedings of the Ninth International Workshop on Machine Learning*.

Scheutz, M., Cantrell, R., and Schermerhorn, P. (2011). Toward Human-Like Task-based Dialogue Processing for HRI. *AI Magazine*.

Schuler, K. (2005). *VerbNet: A Broad-Coverage Comprehensive Verb Lexicon*. PhD thesis, University of Pennsilvania.

Siskind, J. (2001). Grounding the Lexical Semantics of Verbs in Visual Perception using Force Dynamics and Event Logic. *Journal of Artificial Intelligence Research*, 15:31--90.

Sleator, D. and Temperley, D. (1991). Parsing English with a link grammar. Technical Report October, Carnegie Mellon University.

Tellex, S., Kollar, T., Dickerson, S., and Walter, M. (2011). Understanding Natural Language Commands for Robotic Navigation and Mobile Manipulation. In *Proceedings of the National Conference on Artificial Intelligence*.

Thomaz, A., Hoffman, G., and Breazeal, C. (2006). Reinforcement Learning with Human Teachers: Understanding How People Want to Teach Robots. In *The 15th IEEE International Symposium on Robot and Human Interactive Communication*, pages 352--357.

Van Lent, M. and Laird, J. (1999). Learning Hierarchical Performance Knowledge by Observation. In *In Proceedings of the 16th International Conference on Machine Learning*.

Winograd, T. (1972). Understanding Natural Language. *Cognitive Psychology*, 3(1):1----191.

A Model of Interaction

Human-agent interaction for learning with instruction can be viewed on a continuum of instructor/agent control. The control is influenced by what knowledge is expected to be acquired through instructive interactions. Learning by demonstration, imitation or by examples focuses on instructor controlled interactions, in which a human instructor provides examples of task performance, object recognition and categorization etc. The agent observes these examples, maps the performance onto its own capabilities and induces knowledge, goals, or reward functions. Such system place the onus of learning completely on the instructor, requiring the instructor to provide good samples from the feature space such that the agent is able to acquire general hypotheses.

At the other end of the continuum, the agent completely controls the interaction. These systems range from impasse driven learning (Laird et al., 1991) for acquisition of lacking knowledge to incorporating human feedback as a reward in a reinforcement learning framework (Maclin and Shavlik, 1996; Thomaz et al., 2006; Goertzel et al., 2008; Knox and Stone, 2010). Such learning mechanisms severely limit the instructor control over instructions and the opportunity of learn by demonstration and examples is lost.

There are advantages to an approach which explores a mixed-initiative instructional dialog. First, an agent that can assume control on occasion can guide its own learning by requesting clarifications, asking for missing information, and correcting instructor's response based on its own understanding of the state. Second, an agent that can relinquish control on occasion, can take advantage of instructor's knowledge of task structuring and goal decomposition. In a mixed control setting, the instructor can ask the agent for information regarding its state and the environment, verify an agent's learning by questioning the agent, and provide corrections.

A.1 Requirements

To learn from interactive instruction, the agent must maintain a representation of state of the interactions with the instructor, while in acts in the environment and then learns from the instructions by leveraging the context they were provided in. Thus, the agent needs a model of task-oriented interactive instruction. The interaction model must support the properties below.

Requirement	**Type**	**Description**
Integrative	**I1**	Integrate various cognitive capabilites, allowing agent to plan over and reason about a combined space
Flexible	**F1**: Initiation	Both instructor and the agent can introduce new topics
	F2: Knowledge	Accomodates communication regarding nouns/adjectives, prepositions, verbs and related questions
Task-oriented	**T1**: Contextual	Captures dialog-context which is useful for comprehending incomplete sentences, resolving referent ambiguities, learning verbs
	T2: Relevant	Instructon-oriented interpretation of instructor utterances, agent can ask task-relevant queries
	T3: Structural	Organizes dialog such that it is useful in task execution, learning

Dialog-class	Events	Dialog	Segments	Stack
get-next-task	dialog-event	Agent: Waiting for the next command...		
verb-command	dialog-event	Human: Store the orange object.		
attribute-q	dialog-event	Agent: What kind of attribute is orange?		
attribute-ans	dialog-event	Human: A color	O11	
	learning-event	Agent: (learns semantic categorization)		
	dialog-event	Agent: I don't see an orange object. Please teach me to recognize one	O12	
	dialog-event	Human: (clicks on the object). This is orange.		
	learning-event	Agent: (learns perceptual symbol + class)		
get-goal	dialog-event	Agent: What is the goal of the action?		
goal-semantics	dialog-event	Human: The goal is the orange object in the pantry.	A1	
relation-q	dialog-event	Agent: I don't know the preposition in.	G12	
		Human: The orange object is in the garbage.	P121	P121: learn preposition *in*
relation-ans	learning-event	Agent: (learns semantic composition of 'in')		
	learning-event	Agent: (learns goal of 'store')		G12: acquire goal for store
	dialog-event	Agent: What action should I take next?		A1: external action store
	dialog-event	Human: Pick up the orange object.	A13	
get-action	action-event	Agent: (picks up the orange triangle)		
verb-command	dialog-event	Agent: What action should I take next?		
	dialog-event	Human: Put the object in the pantry.	A14	
	action-event	Agent: (puts the triangle in the pantry)		
	dialog-event	Agent: What action should I take next?	A15	
	dialog-event	Human: You are done.		
	learning-event	Agent: (learns 'store' with retrospective projection)		

Figure 6: Annotated Human-Agent Dialog for Acquisition of *store*

A.2 Formulation

The interaction model we have developmental is based on a theory of discourse structure that stresses the role of purpose in discourse (Grosz and Sidner, 1986). Our implementation has been adapted from Rich and Sidner (1998) by extending the framework to accommodate learning from collaborative situated interaction. To describe how the state of interaction is represented, we define the following:

A.2.1 Event

An event causes change in the environment state (*action-event*), in the discourse state of the instructor-agent interaction (*dialog-event*), or in agent's knowledge (*learning-event*). Action-events correspond to various primitive and composite actions that can be executed in the environment. Instructor and agent utterances are categorized dialog-events. Dialog-events can be assigned to different classes based on their lexical and syntactic structures. Some relevant dialog-event classes and examples are shown in Figure 6 (dialog-class column). A learning-event is successful acquisition of linguistic mapping, semantic, or procedural knowledge. Our model accomodates for action-, dialog- learning-events for the agent but only for dialog- for the instructor.

All events are handled in a similar fashion for determining the state of interaction, addressing the *Integrative* (I1) requirement of interaction model design. Figure 6 shows an example of human-agent dialog annotated with different events that occur in a task-oriented interaction.

A.2.2 Segment

A discourse segment is a contiguous sequence of events that serve a specific *purpose*. This linguistic structure serves to organize a dialog in purpose-oriented chunks enabling the agent to reason about interactions and set corresponding goals for behavior (*task-oriented* requirement T3). Figure 6 shows how human-agent communication is organized in segments. Segment `A1` begin when the instructor asks the agent to perform an action in the environment. Since the agent does not know how to execute this action, it prompts the instructor to provide several pieces of information related to the action including information about objects (segment `O11`), spatial prepositions (segment `P121`) and the goal of the verb (segment `G12`). Each of these segments end when the agent performs the required learning. A segment is characterized by the following:

- `purpose-set`, a set of events that signify that the purpose of the segment is achieved. The agent uses some pre-encoded *instructional-learning-based* heuristics (task-oriented requirement `T2`) to assign purpose-set to segments. These heuristics are motivated by the fact that the agent is acting in a collaborative environment and the agent's intentions closely follow the instructor's intentions. For the example dialog shown in Figure 6, all segments beginning with action-commands (`A1`, `A13`, `A14`) have an associated purpose of external action execution. Similarly, all question segments (`O11`, `G12`) seek to establish shared belief about the environment and expect a dialog-event from the instructor in response. The purpose-set of the segments influences agent's decision making.
- `satisfaction-set`, a set of events that are indicative that the purpose of the segment has been achieved and signify that the segment has ended. This usually is a superset of the purpose-set, since if an event from the purpose set occurs, the purpose of the segment is over and the agent can stop tracking the related goal. The satisfaction-set is also determined using pre-encoded *instructional-learning-based* heuristics.
- `context` captures why the segment was created (*task-oriented* requirement, `T1`). This context useful in parsing and interpretation of incomplete sentences and dialog-based noun-phrase reference resolution. In Figure 6 segment `O11`, the fact that the noun-phrase fragment *a color* was provided in response to the question, is useful in parsing the fragment and learning that *orange is a color*.

A.2.3 Stack

The attentional structure of discourse is captured by the focus stack of segments. As interaction between the instructor and the agent progresses, and as agent performs actions and acquires knowledge, new segments are created, pushed onto the focus-stack and popped off. The segments that are on the stack are termed open segments and the segment at the top represents the current focus of the interaction and the agent acts to achieve the corresponding purpose. The interaction state at a time instance is shown in Figure 6. The stack contains three open segments (`P121`, `G12`, `A1`) that are hierarchically connected, each segment contributing towards the purpose of its parent. To be able to learn *move*, the agent must know its goal and to comprehend the goal, the agent must know the spatial concept corresponding to the preposition *in*.