Learning Grounded Language through Situated Interactive Instruction

Shiwali Mohan*, Aaron Mininger*, James Kirk*, John E. Laird
Computer Science and Engineering
University of Michigan, Ann Arbor, MI 48109-2121
{mohan, mininger, jrkirk, laird}@umich.edu

Abstract

We present an approach for learning grounded language from mixed-initiative human-robot interaction. Prior work on learning from human instruction has concentrated on acquisition of task-execution knowledge from domain-specific language. In this work, we demonstrate acquisition of linguistic, semantic, perceptual, and procedural knowledge from mixed-initiative, natural language dialog. Our approach has been instantiated in a cognitive architecture, Soar, and has been deployed on a table-top robotic arm capable of picking up small objects. A preliminary analysis verifies the ability of the robot to acquire diverse knowledge from human-robot interaction.

Introduction

Roy (2005) defines *language grounding* as a process for relating words and speech acts to a language user's environment via grounded beliefs. The ability to associate language with objects, events, and actions allows language users to coordinate effort on collaborative tasks, establish shared beliefs about the environment, and learn from others' experiences through linguistic communication. An embodied, artificial agent that can effectively collaborate with humans should be able to comprehend language by connecting linguistic symbols to its perceptions, actions, experiences, and learning.

We are investigating mechanisms through which embodied, cognitive agents can acquire grounded representation of natural language. We are interested in developing robotic agents that can learn to associate language to various aspects of cognition (perception and spatial, semantic, and procedural knowledge) that may originate outside of the linguistic system. In this paper, we focus on learning three categories of linguistic symbols: adjectives/nouns (*large*, *orange*, *cylinder*) that describe perceptual features (color, shape, size) of simple objects (foam blocks), prepositional structures for spatial relations (*left of*, *in front of*), and verbs (*move)* that are compositions of primitive robotic actions (*point*, *pick up*, *put down*).

One of the most efficient means of learning is through interactive instruction provided by a human expert. The instructor can provide real-world examples of novel nouns, adjectives, prepositions, and verbs accompanied with corresponding linguistic symbols. Through a grounding process, the agent comprehends the utterance by connecting it to physical objects it perceives and actions it performs. The agent then encodes and stores the mapping from linguistic symbols to non-linguistic knowledge extracted from the real-world examples. This linguistic knowledge along with the real-world context can be leveraged to acquire more complex linguistic and non-linguistic knowledge. The instruction approach we are pursuing has the following characteristics:

1. **Situated:** Instructions are interpreted within a real-world context, eliminating many forms of ambiguity. The agent grounds the words in the instructor's utterance to objects, spatial relationships, and actions. This process allows the agent to extract specific examples from real-world scenarios, which form the basis of learning. The knowledge so acquired is directly relevant to the tasks the agent is required to perform.
2. **General:** The instruction mechanism can be applied to any type of missing knowledge, including object identification and categorization, labeling objects with words, learning action models, or labeling actions.
3. **Interactive:** We focus on mixed-initiative interaction and bi-directional flow of information. When the agent's knowledge is insufficient for it to make progress on a task, or the instructions are incomplete or ambiguous, it can ask for clarifications. 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.
4. **Knowledge-level interaction:** The instructor provides knowledge to the agent by referring to objects and actions in the world, not the agent's internal data structures and processes.

In addition to focusing on instruction, our approach is to develop agents using a cognitive architecture (Langley, Laird, Rogers 2009) that is integrated with perceptual and motor systems. The cognitive architecture provides task-independent knowledge representations; memories to hold both short-term and long-term knowledge; processes for decision making, accessing memory, and learning; and interfaces that support the transductions of continuous perception into symbolic structures and discrete motor commands into continuous control systems. The same

* The first three authors contributed equally to this work.

architecture is used across all tasks, as is task-independent knowledge that supports general capabilities. Domain knowledge specializes behavior to specific tasks so that a single agent that is encoded with knowledge for many domains can pursue a variety of tasks. Through learning an agent can dynamically extend its task knowledge.

The cognitive architecture we are using is Soar (Laird 2012), which has been in development and use for over 30 years and has been applied to a wide variety of domains and tasks, including natural language understanding and robot control (Laird et al. 1991; Laird & Rosenbloom 2000; Benjamin, Lonsdale, & Lyons 2006). Recently, we have made significant extensions to Soar, including the addition of episodic and semantic memories, as well as a visual-spatial system, that enhance Soar's ability to support grounded language learning.

In the following section, we discuss the related work and describe our system and its environment. We then describe Soar, emphasizing the components that implement the instruction process, and those that hold knowledge learned through instruction. We then give an overview of the instruction process. This is followed by more detailed descriptions of how the instruction process is applied to noun/adjective, preposition, and verb learning.

Related Work

Much of the prior research on learning from human interaction has focused on learning from demonstration, imitation, and observation (Argall et al., 2009). Although these methods can be useful, they are not efficient for communicating the hierarchical goal structures that are often needed for complex behavior. By using human-directed instruction, we eliminate the need to perform activity (and goal) recognition, as well as potentially complex mappings between different reference frames. Another difference is that our agent demonstrates its understanding through performance – it can use what it has learned to demonstrate new behaviors.

Recent work by Allen et al. (2007) demonstrates a collaborative task learning agent that acquires procedural knowledge through a collaborative session of demonstration, learning, and dialog. The human teacher provides a set of tutorial instructions accompanied with related demonstrations in a shared environment. The agent uses these situated examples to acquire new procedural knowledge. Although the learning is human demonstration driven, the agent controls certain aspects of its learning by making generalizations without requiring the human to provide a large number of examples. A key distinction of our work is that the initiative of learning is placed with both the instructor and the agent. The agent can initiate a learning interaction if its knowledge is insufficient for further progress, and the instructor can verify the agent's knowledge by asking relevant questions.

Chen et al. (2010) describe a unified agent architecture for human-robot collaboration that combines natural language processing and common sense reasoning. The agent is essentially 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. Our agent is able to use interactive instruction to learn a wider variety of knowledge, including grounded representation of language.

A significant line of research for grounded language acquisition is being pursued by Tellex et al. (2011). Their framework instantiates a probabilistic graphical model for a particular natural language command according to the command's hierarchical and compositional semantic structure. They focus on corpus based learning mechanisms, while we concentrate our efforts on interactive learning where the agent learns from a two way communication with the human instructor.

System Description

In order to integrate our system into a real world domain, we have used a simple table-top environment (shown in Figure 1) with a robot arm[1]. The domain simulates a toy kitchen; it includes three locations – a *stove*, a *dishwasher*, and a *pantry*. It also consists of a variety of movable objects (foam blocks) of different colors, sizes and shapes.

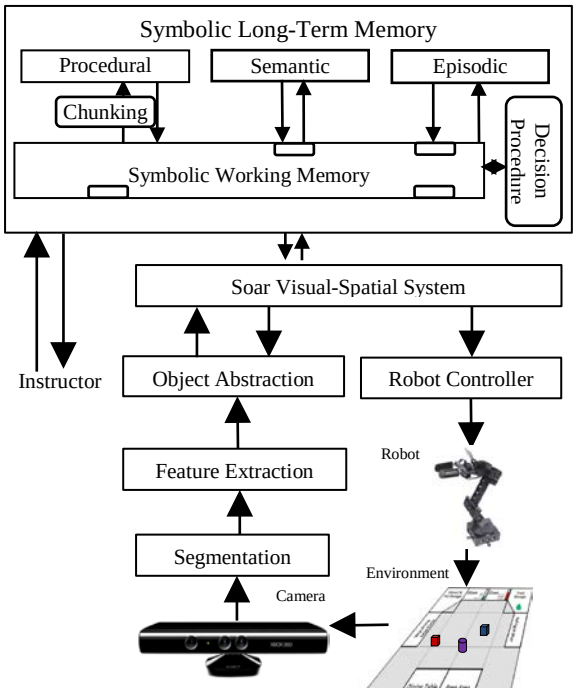

Figure 1 System Description

Perception System

The perception system segments the scene into objects using both color and depth information provided by an overhead Kinect camera. Features useful for distinguishing colors, shapes, and sizes are extracted for each object found in the scene. A symbolic description of each object

[1] The robotic arm and visual system were developed by the April lab at the University of Michigan headed by Edwin Olson.

is constructed by determining the color, shape, and size of the object from the corresponding features using a K-Nearest Neighbor algorithm**.** This description, along with position and bounding box information, is sent to the agent. Similar representations are created for the locations.

Robot Controller

To act in the world, the agent sends high-level arm commands to the robot controller which calculates and performs the low-level motor commands. The robot is capable of executing the following object manipulation actions: `point-to <o>, pick-up <o>, put-down <x,y>.`

Instructor Interface

The instructor interacts with the agent through a simple chat interface. Messages from the agent are converted to natural language using templates. Messages to the agent are fed through the link-grammar (Sleator and Temperley, 1993) parser to extract part of speech tags and sentence structure. The instructor can also 'point' to an object by clicking it on a live camera feed. Information about the selected object is provided to the agent as a perception and is used to resolve references to the word *this*.

Basics of Soar Cognitive Architecture

Figure 1 shows the structural view of the Soar cognitive architecture including its primitive memories, learning mechanisms, and decision processes.

Soar Visual-Spatial System

Soar contains a task-independent Spatial Visual System (SVS) that supports interaction 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 evaluated 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 in the architecture, but predicate extraction is done using task-specific knowledge.

Working Memory

On the symbolic side, working memory maintains relational representations of current and recent sensory data, current goals, and the agent's interpretation of the situation given its goals. Working memory buffers provide interfaces to Soar's long-term memories and the perceptual system, robot controller, and instructor interface.

Semantic Memory

Soar contains a long-term semantic memory which is used to store and retrieve declarative facts about the world and structural regularities of the environment. This memory is context independent; it contains knowledge that is not related to when and where it was acquired. The agent can deliberately store parts of its working memory into semantic memory as concepts. It can deliberately retrieve a concept using a cue created in a special working memory buffer. The best match in semantic memory (biased by recency and frequency) is then added to working memory. In our agent, semantic memory stores the linguistic knowledge along with facts about the domain.

Episodic Memory

Soar's episodic memory is a context dependent long-term memory that records the agent's. It effectively takes snapshots of working memory (episodes) and stores them in a chronological fashion, enabling the agent to remember both the context and temporal relations of past experiences. The agent can deliberately retrieve an episode by creating a cue in a special working memory buffer. The best partial match in episodic memory (biased by recency) is then added to working memory. Soar's episodic memory facilitates retrospective learning through situated instruction by automatically encoding and storing all sensory information. The agent can review past instructions and observe the resulting changes in both the environment and internal state.

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 deliberate 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.

Chunking is a learning mechanism that creates rules from the reasoning that occurred in a substate. When a result is created in a substate, a production is created whose conditions are the state descriptors that existed before the substate and were tested in it, and whose actions are the result. This production is added to long-term memory and is immediately available.

Situated Interactive Instruction

Situated interactions between the agent and the instructor can be initiated by either party. Instructor-initiated interactions include asking the agent to execute a certain

behavior ("*pick up the red triangle*"), basic queries ("*which is the green* block"), and demonstrative sentences ("*The red block is to the right of the yellow triangle*"). The agent can react by performing the required action, responding to queries by generating a natural language response, or acquiring knowledge from demonstrative sentences.

The interaction, behavior execution, and knowledge acquisition in the agent are tightly integrated. If the agent lacks knowledge to comprehend an utterance or execute an action, it engages the instructor in a dialog through which it attempts to acquire the missing piece of knowledge. The learning mechanisms for noun/adjectives, prepositions, and verbs are described in later sections; in this section we focus on the interaction model for instructional learning.

Interaction Model

One of the challenges of interactive instruction is that the agent must maintain a representation of the state of interactions with the instructor while acting in the environment, and then learn from the instructions in the context they were provided in. Thus, the agent needs a model of task-oriented interaction. If required, the context can be recreated at a later stage using episodic memory. The interaction model is required to support the properties described below.

1. Both the instructor and the agent can assume control of the interactions at any time.
2. The interaction model provides a context for instructor's elicitation, allowing the agent to take relevant actions.
3. The interactions by the agent should be informed by agent's reasoning, learning, and acting mechanisms.
4. The interaction model and the sequence of interactions should inform the agent's learning.

The interaction model we use has been adapted from Rich and Sidner (1998). It captures the state of task-oriented interaction between the agent and the instructor. To formalize the state of interaction, we introduce (1) *events* that change the state of interaction, (2) *segments* that establish a relationship between contiguous events, and (3) a *focus-stack* that represents the current state of interaction. Possible events include dialog utterances, actions, and learning events. The dialog-events can originate from the agent or the human instructor. Actions and learning events are constrained to the agent.

In accordance with the discourse interpretation algorithm described by Rich and Sidner (1998), each event changes the focus-stack by either (i) starting a new segment whose purpose contributes to the current purpose (pushing a new segment with a related purpose on the focus stack), (ii) continuing the current segment by contributing to the current purpose, (iii) completing the current purpose (eventually popping the focus stack), or (iv) starting a new segment whose purpose does not contribute to the current purpose (pushing a new, interrupting segment on the focus-stack and changing the purpose of the interaction). Every instructor dialog utterance is processed as described below:

1. Syntactic Parsing: A syntactic parse is generated based on a static dictionary and grammar using the link-grammar parser. We have incorporated LG-Soar[2] (Lonsdale et al., 2006), a natural language component implemented as productions in Soar, into our agent. It uses POS tags to create a condensed structure identifying the useful content of the message. This static parse of the sentence is categorized further as an *action-command*, *goal-description*, *descriptive-sentence*, etc.

2. Intention Association: Based on the categorization of the sentence and the information contained in it, the agent associates an intention (purpose) with the sentence. For example, through an *action-command*, the instructor intends for the agent to perform an action in the environment. Currently, intentions are heuristically derived. These heuristics are motivated by the domain characteristics. Although significant work exists in the area of intention recognition, this has not been the focus of our work. We find that simple heuristics are useful for the learning scenarios we are interested in.

The focus-stack is updated based on the intentions associated with the instructor's utterance and the current state of the focus-stack. If the instructor utterance satisfies the purpose of an open segment on the focus stack, it is processed and the segment is terminated. Otherwise, an appropriate segment is initiated on the focus-stack.

3. Grounded Comprehension: In the next phase, the agent attempts to associate the words in the sentence to its perceptual, spatial, semantic, and procedural knowledge.

The noun phrases are mapped to either the objects present in the perceptual field. The noun phrase in a sentence such as *a blue cube* or *the dishwasher* is a linguistic description of an object present in the environment. Given a mapping between linguistic and perceptual symbols, the noun phrase can be mapped to a perceptual description of the object.

Prepositions are mapped to spatial relationships which are composed of primitive spatial relationships that are known to the agent. In the sentence, "*The red object is right of the blue object.*" the prepositional phrase *right of* is mapped to a spatial symbol representing the learned spatial relation between the two objects.

Verbs are mapped to actions known to the agent through linguistic-procedural mappings stored in agent's semantic memory. After a mapping has been obtained, the action is instantiated with objects present in the perceptual field or semantic memory based on the noun phrases used as arguments of the verb.

4. Behavior Execution: If phase 3 is successful and the agent is able to determine the grounding of the entire sentence, it performs a behavior based on the intention associated with the instructor's utterance. If the instructor asks "*Which is the blue block?*", the agent may perform the action `point-to obj12`, where `obj12` is an object that satisfies the description of *the blue block*. A command

[2] Adapted by Sam Wintermute at Soar Technology.

"*Pick up the red triangle*" results in the agent picking up the required block.

5. Learning: A lack of knowledge to progress further in Soar results in an impasse. Our agent resolves an impasse by initiating an interaction with the instructor about the knowledge it lacks. This interaction provides the agent with situated examples, which are used to derive and acquire generalized knowledge. Learning can potentially occur during every phase previously described, i.e. if the agent fails at generating a syntactic parse of the instructor's utterance, it can reach a resolution by communicating with the instructor. Similarly, if the intention of the instructor is ambiguous, further communication can aid in correct intention assignment.

In this paper, we focus on impasses arising during the grounded comprehension and the behavior execution phase. Impasses arise in grounded comprehension because the agent lacks mapping knowledge that associates words with perceptual information or knowledge of the domain. The impasse driven interaction during this phase allows the agent to acquire linguistic-perceptual symbol mappings (discussed in detail in Noun and Adjectives), linguistic-spatial knowledge mappings (in Prepositions), and linguistic-action mappings (in Verbs). Interactions initiated to resolve impasses in the behavior execution phase allow the agent to acquire action execution knowledge (discussed further in Verbs).

Nouns and Adjectives: Perceptual Symbols

Throughout its execution, the agent learns new nouns and adjectives which help it better understand the objects it can perceive. This knowledge is used to facilitate communication between the instructor and the agent by building a richer descriptive vocabulary. Currently we limit ourselves to two types of nouns: those used as visual adjectives (e.g. *red* or *triangle*) and the categories of those adjectives (e.g. *color* or *shape*). Three types of learning are currently supported: semantic (an adjective associated to its category), linguistic (an adjective associated with its class label), and perceptual (a set of features associated with a class label).

Background Knowledge

Perceptual: The visual system has pre-coded knowledge about how to extract useful features for each of the three visual categories: *color*, *shape*, and *size*. For example, the average red, green, and blue values taken across all the pixels of a segmented object are used as color features during classification.

Semantic: The agent knows about the three visual categories, but does not start with any semantic knowledge about specific adjectives like *red* or *triangle*.

Noun/Adjective Acquisition

Semantic: During its execution the agent builds up a mapping in semantic memory from an adjective to its category (*red* to *color* or *triangle* to *shape*). The addition of a new mapping can be initiated by the instructor with an instruction like "*Red is a color*." The agent can also ask the instructor about the meaning of an adjective it doesn't know, e.g. "*To what category does red belong?*" The construction of this mapping can thus be both agent-initiated and instructor-initiated.

Linguistic and Perceptual: Linguistic knowledge about adjectives is built-up in a mapping from an adjective to a class label used by the visual system. When the agent learns a new adjective and its category, it instructs the visual system to create a new class within that category and gives it a new label (e.g. create a new class within the color classifier and label it *color14*). We add this extra layer of indirection in anticipation of future work where an adjective might have multiple meanings and a disambiguation step would be required. Every time the instructor refers to an object using that adjective, the visual system uses that object as a training example for the associated class label. This refines the perceptual knowledge of which features correspond to a class label. Through experience and interaction the agent improves its ability to recognize and distinguish between those labels.

Prepositions: Spatial Relations

Spatial relationships describe the location and/or orientation of an object in space with respect to another object. These relationships, such as *near* or *right-of*, provide useful descriptive information about the current world. In this paper, we initially limit ourselves to learning spatial relationships that are based on alignment and not distance or orientation.

The learning of spatial relationships and their associated prepositions depends on spatial, semantic, and linguistic knowledge. Spatial knowledge, in the form of primitive spatial relationships, is built into the system, while the composition of spatial primitives (semantic knowledge) and the mapping of the prepositional phrase to this composition (linguistic knowledge) are learned.

Background Knowledge

Spatial: The learning of new spatial relations depends on a set of spatial primitives that are built into the Spatial Visual System. The built in primitives consist of three basic relations based on alignment for each axis: *x*, *y*, and *z*. These basic relations are *aligned*, *greater-than*, or *less-than*. These can be composed to describe a large body of complex relations. For example, the relationship *intersecting* can be described by an alignment of all three axes. SVS can be queried for the truth value of all the possible primitive relationship between any two objects.

Preposition Acquisition

Semantic: The user teaches the agent a new spatial relation by referring to two objects in the world that demonstrate that relationship. SVS is queried for all of the current true primitive relationships between the specified

objects. The learned spatial relationship is represented by a composition of all the true primitives extracted from a situated example. When teaching "Object a is right of object b", the true primitives between 'a' and 'b' could be *Y–aligned, X-greater-than,* and *Z-aligned.*

This mechanism enables the learning of a new preposition with just one example. However it is possible that the learned information is more specific than intended. For example, the Y-alignment could be incidental and not an intended requirement for *right-of.* Additional training examples, where the objects are not aligned along the Y-axis, will correct the initial assumptions and generalize the learned spatial relation.

Linguistic: The prepositional phrase used to describe the new relationship is mapped to the learned composition. In the above example, *right-of* is mapped to this composition and stored into semantic memory (shown in Figure 2), where it can be accessed when querying about the spatial relationships between other objects in the future.

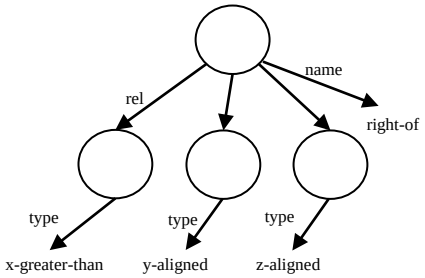

Figure 2 Preposition: Spatial Relation Semantic Mapping

Verbs: Actions

A verb is a word that conveys an action (*walk, run*), an event (*occur, happen*), or a state (*be, exist*). In this paper, we limit ourselves to action verbs. Specifically, we are interested in actions that have declarative, perceptible goals in the world. To learn a new verb, the agent attempts to acquire three pieces of knowledge: linguistic knowledge (the corresponding word and its argument structure), semantic knowledge (knowledge of the goal of the verb), and procedural knowledge (the action execution knowledge to achieve the goal corresponding to the verb).

Consider the verb *move* and its usage, "*Move the red block to the pantry*." The verb has two arguments, the object to be moved – *the red block*, and the location the object should be moved to – *the pantry*. This particular instance of *move* corresponds to a specific action instantiated with the indicated arguments. The action is a composition of two known primitives, *pick up the red block*, and *put the block in the pantry*. The action is successful when the correct object is placed in the intended location. To learn the verb *move*, the agent acquires the argument structure of the verb (linguistic knowledge), the correct composition of primitives (procedural knowledge) and the goal of the action (semantic knowledge).

Background Knowledge

Linguistic: To be able to comprehend verbs and execute them in the environment, the agent needs to associate the verb and its argument structure to actions and objects in the environment. This mapping is encoded declaratively in the agent's semantic memory and allows the agent to access the related action operator instantiated with objects in the environment.

Consider the example shown in Figure 3. It maps the verb *put* with an argument structure consisting of a *direct object* and the object connected to the verb via the preposition *in* with the operator `op_put-down-object-location`. This allows the agent to associate the sentence "*Put a red, large block in the dishwasher*" with an appropriate operator which will achieve the intended goal.

Procedural: The agent has pre-programmed rules that allow it to execute the primitive actions in the environment. The actions are implemented through operators in Soar. An action is defined by its *availability conditions* (the preconditions of the action), *execution knowledge* (rules that execute action commands in the environment), and *termination conditions* (a set of predicates that signify that the goal of the action has been achieved). The agent generates the set of available primitive actions based on its current perceptions. This set includes all the actions the agent can take given the current physical constraints, object affordances, and the agent's domain knowledge. The agent also contains domain action models (encoded as productions) with which it can simulate the effect of its actions on the environment during learning.

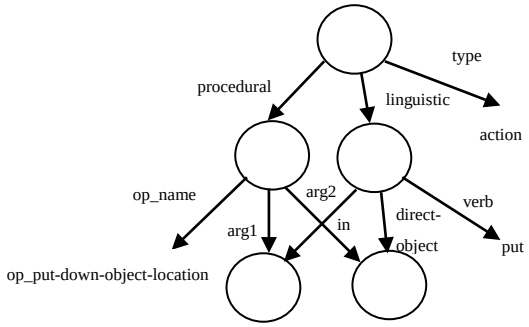

Figure 3 Verb: Action Mapping

Verb Acquisition

Acquisition of new verbs is integrated with interactive execution of tasks. If the agent cannot comprehend a verb in the instructor's action commands, i.e. it is unable to map the verb to a known action; it tries to learn it through situated interactions. In particular, it learns the following knowledge.

Linguistic: Using the argument structure of the action command (extracted by the syntactic parser), the agent creates a new mapping (similar to Figure 3) in its sematic memory. This mapping associates the novel verb and its argument structure to a new operator and its arguments.

Semantic and Procedural: After the agent creates a new verb-operator indexical mapping for the novel word, it

proposes the new operator in an attempt to execute the action. Since it is a new action, the agent does not possess the execution rules of this action. The agent begins an interaction with the human instructor to acquire a situated example execution of the action. Through this interaction, the instructor decomposes the action into a sequence of primitive actions known to the agent, which the agent executes in the environment. Note that these interactions are automatically stored in the agent's episodic memory and are later used for retrospective learning. When the agent successfully completes the instructed execution of the verb, it queries the instructor for a declarative goal description, which it encodes in its semantic memory.

Once the agent acquires the goal of the verb, it attempts to learn the conditions under which it should execute the instructed operators to achieve the verb's goal. This process involves internally projecting the instructed operators starting from the state the agent was in when the new verb was first suggested (Huffman and Laird, 1995). The internal projection generates a causal explanation from which the agent learns general procedural knowledge for the new verb immediately after its first execution. The individual steps are described below.

1. State Reconstruction: In a substate, the agent queries its episodic memory for the episode in which it was first asked to execute the new verb it is attempting to learn. As the required episode is retrieved, the agent copies the relevant state attributes to the substate. The agent queries its semantic memory for the goal of the verb, which is copied to the current substate as the desired state. The agent then proposes the operator corresponding to the new verb. Since it has not yet learned the execution rules for this new operator, it experiences an impasse.

2. **Retrospective Projection:** The next step is to project the effect of instructed primitive actions in the reconstructed state using domain action models. The agent queries its episodic memory for an episode containing the next primitive action provided by the instructor. It then applies this action-command to the current state and advances the state using its action models. This process is continued until the desired state (identified in the reconstruction step) is achieved.

3. Explanation-based Generalization: If the projections of instructed actions are successful in achieving the intended goal state of the instruction, Soar's chunking mechanism compiles the reasoning about projections into rules that incorporate tests for why each operator had to be selected to achieve the verb's goal. These rules are added to agent's procedural knowledge and are immediately available to the agent for preforming the verbs actions.

Case Study

To evaluate the linguistic, interaction and learning capabilities of our agent, we presented the robot with a composite action command, "*Move the red triangle to the pantry*". Note that this was presented in the first interaction with the agent. At this time point, the agent's knowledge is limited to the background knowledge described in previous sections. It does not have grounded representations for the verb *'move'*, the adjective *'red'*, the noun *'triangle'* and the preposition *'in'*. The agent will acquire this knowledge through various interactions with the instructor.

LG-Soar categorizes this command as an action-command and the agent attempts to generate a grounded instantiation of an action corresponding to this command. It first attempts to ground all the noun-phrases (NPs) to objects and location present in the environment. The NP *'the pantry'* is successfully mapped to the correct location. However, grounded comprehension of *'the red triangle'* fails, since the agent does not know how the words *'red'* and *'triangle'* map to its sensory information. Through questions such as "*To what category does red belong?*" and "*To what category does triangle belong?*", the agent prompts the instructor for categorization information for these words. The instructor answers with *'a color'* and *'a shape'*. Then the agent prompts the instructor for a specific example of *'a red triangle'* from the current scene. The instructor points to the intended object. The agent then generates perceptual symbols for the color *'red'* and shape *'triangle'*, which are provided to the visual system for training. With a only a single example, the agent learns to correctly detect a red triangle from the scene. The NP *'a red triangle'* is eventually mapped to an object.

The agent then attempts to instantiate an action corresponding to the verb *'move'*. It queries its semantic memory for an operator that corresponds to the verb 'move' and takes two arguments. The retrieval failure causes the agent to create a new mapping to a new operator `op-1`. This operator is proposed. Since, the agent does not know how to apply this operator, it begins an instructed trial. The agent informs the instructor that it cannot progress further and asks the instructor, "*What next action should I take?"* The instructor can now decompose the task into primitive actions which will later be composed as application knowledge for the operator `op-1`. The instructor tells the agent to *'pick up the red triangle'*, which is successfully mapped to the `pick-up` operator and applied. The agent again prompts the instructor for the next action to execute. The instructor replies with *'put the object in the pantry'*.

Since, the agent does not have the grounded representation for the preposition *'in'*, it begins an interaction for learning this representation by asking the instructor, *"I do not know the spatial relationship in, please teach me with examples"*. The instructor can provide situated examples from the scene, "*the green square is in the dishwasher"*. The agent extracts the spatial predicates from this examples and learns a mapping for *'in'*. The agent then proceeds execute the `put-down` action.

The instructor then indicates that the agent has completed *'move'*. The agent then learns this composition through retrospective projection. The application knowledge acquired for the verb 'move' is general, and the agent can perform "*move the yellow square to the dishwasher"* without further training, provided it can ground the NP *"the yellow square"*.

Summary and Discussion

Grounded language acquisition can be described as learning to associate linguistic forms (words) to knowledge about the environment. This knowledge can be perceptual, spatial, semantic, or procedural and may or may not be acquired through linguistic interaction. The ability to ground language by associating words to objects, actions, and knowledge allows the agent to comprehend the instructor's utterance and collaborate with the human.

In this paper, we discussed an approach for acquisition of grounded language through situated, interactive, instruction. The agent is able to learn new nouns and adjectives, prepositions, and verbs by using situated examples provided by the instructor in a shared environment. Along with acquiring mappings for novel words, the agent also acquires new perceptual, semantic, and procedural knowledge which provides the grounding for the corresponding words. The spatial knowledge is composed of task-independent primitives and is not learned from interactive instruction. The agent can use the acquired knowledge for its own reasoning, including learning relevant properties of objects, spatial relations, and new abstract actions it can use in planning.

The agent learning is incremental; acquisition of noun/adjective mapping allows the agent to extract situated examples of spatial relationships from the utterance "*The red triangle is to the right of the yellow block*" and the corresponding visual scene. Using this situated example, the agent not only learns the grounded meaning of *right of*, but can also use this knowledge to learn more complex actions.

Future Work

There are several avenues of that we are interested in exploring. The primary focus will be on extending the learning capability of the agent. The current work involving objects has been limited to acquiring adjectives and nouns that correspond to visual features of the objects. The next step is acquiring knowledge for perceptual and functional categorization and classification. We expect the agent to be able to acquire knowledge such as cans are gray cylinders, or that only objects of a certain size can be picked up. This knowledge could aid in acquiring and generalizing action affordances.

Currently, the system is able to acquire binary spatial relationships such as *right-of*. We plan to extend the system such that the agent is able to acquire complex relationships involving multiple objects, such as *between*. Significant work will involve using the definitions of newly acquired prepositions for predicate projections and integrating prepositions with action execution, allowing the agent to comprehend and execute complex action commands such as, "*Put the red cube between the yellow cylinder and the large triangle*".

Under the current constraints, the agent can acquire action verbs that are compositions of known primitives and have perceptible goals. The goals have to be communicated declaratively to the agent. Our future efforts will focus on acquiring goals autonomously from multiple executions of the verb in the environment. We are also interested in expanding the kinds of verbs the agent can learn including non-action verbs.

Acknowledgments

The work described here was supported in part by the Defense Advanced Research Projects Agency under contract HR0011-11-C-0142. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressly or implied, of the DARPA or the U.S. Government.

References

Allen, J.; Chambers, N.; Ferguson, G.; Galescu, L.; Jung, H.; Swift, M.; and Taysom, W. 2007. Demonstration of PLOW: A Dialogue System for One-Shot Task Learning. In Proceedings of Human Language Technologies.

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

Chen, X., Ji, J., Jiang, J., Jin, G., Wang, F., Xie, J. (2010) Developing High-level Cognitive Functions for Service Robots, *In Proceedings of 9th International Conference on Autonomous Agents and Multi-agent Systems.*

Huffman, S. B., and Laird. J. E. (1995). Flexibly Instructable Agents, *Journal of Artificial Intelligence Research*, 3, 271-324.

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

Laird, J. E., Hucka, M., Yager, E., and Tuck, C. (1991). Robo-Soar: An integration of external interaction, planning, and learning, using Soar, *IEEE Robotics and Autonomous Systems*. 8(1-2), 113-129.

Laird, J. E., and Rosenbloom, P. S., (2000). Integrating Execution, Planning, and Learning in Soar for External Environments, *Proceedings of the National Conference of Artificial Intelligence.*

Langley, P., Laird, J. E., and Rogers, S. (2009). Cognitive Architectures: Research Issues and Challenges, *Cognitive Systems Research*, 10(2), 141-160.

Lonsdale, D., Tustison, C., Parker, C., and Embley, D. W. (2006) Formulating Queries for Assessing Clinical Trial Eligibility, *Proceedings of the 11th International Conference on Applications of Natural Language to Information Systems*.

Rich, C., and Sidner, C. 1998. COLLAGEN: A Collaboration Manager for Software Interface Agents. *User Modeling and User- Adapted Interaction*.

Roy, D. (2005). Semiotic Schemas: A Framework for Grounding Language in Action and Perception. *Artificial Intelligence*.

Sleator, D., and Temperley, D. 1993. Parsing English with a Link Grammar. *Third International Workshop on Parsing Technologies.*

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