Interactive Learning of Grounded Verb Semantics towards Human-Robot Communication

Lanbo She and **Joyce Y. Chai**
Department of Computer Science and Engineering
Michigan State University
East Lansing, Michigan 48824, USA
{shelanbo, jchai}@cse.msu.edu

Abstract

To enable human-robot communication and collaboration, previous works represent grounded verb semantics as the potential change of state to the physical world caused by these verbs. Grounded verb semantics are acquired mainly based on the parallel data of the use of a verb phrase and its corresponding sequences of primitive actions demonstrated by humans. The rich interaction between teachers and students that is considered important in learning new skills has not yet been explored. To address this limitation, this paper presents a new interactive learning approach that allows robots to proactively engage in interaction with human partners by asking good questions to learn models for grounded verb semantics. The proposed approach uses reinforcement learning to allow the robot to acquire an optimal policy for its question-asking behaviors by maximizing the long-term reward. Our empirical results have shown that the interactive learning approach leads to more reliable models for grounded verb semantics, especially in the noisy environment which is full of uncertainties. Compared to previous work, the models acquired from interactive learning result in a 48% to 145% performance gain when applied in new situations.

1 Introduction

In communication with cognitive robots, one of the challenges is that robots do not have sufficient linguistic or world knowledge as humans do. For example, if a human asks a robot to *boil the water* but the robot has no knowledge what this verb phrase means and how this verb phrase relates to its own actuator, the robot will not be able to execute this command. Thus it is important for robots to continuously learn the meanings of new verbs and how the verbs are grounded to its underlying action representations from its human partners.

To support learning of grounded verb semantics, previous works (She et al., 2014; Misra et al., 2015; She and Chai, 2016) rely on multiple instances of human demonstrations of corresponding actions. From these demonstrations, robots capture the state change of the environment caused by the actions and represent verb semantics as the desired goal state. One advantage of such state-based representation is that, when robots encounter the same verbs/commands in a new situation, the desired goal state will trigger the action planner to automatically plan a sequence of primitive actions to execute the command.

While the state-based verb semantics provides an important link to connect verbs to the robot's actuator, previous works also present several limitations. First of all, previous approaches were developed under the assumption of perfect perception of the environment (She et al., 2014; Misra et al., 2015; She and Chai, 2016). However, this assumption does not hold in real-world situated interaction. The robot's representation of the environment is often incomplete and error-prone due to its limited sensing capabilities. Thus it is not clear whether previous approaches can scale up to handle noisy and incomplete environment.

Second, most previous works rely on multiple demonstration examples to acquire grounded verb models. Each demonstration is simply a sequence of primitive actions associated with a verb. No other type of interaction between humans and robots is explored. Previous cognitive studies (Bransford et al., 2000) on how people learn have shown that social interaction (e.g., conver-

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 1634–1644
Vancouver, Canada, July 30 - August 4, 2017. ©2017 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17-1150

sation with teachers) can enhance student learning experience and improve learning outcomes. For robotic learning, previous work (Cakmak and Thomaz, 2012) has also demonstrated the necessity of question answering in the learning process. Thus, in our view, interactive learning beyond demonstration of primitive actions should play a vital role in the robot's acquisition of more reliable models of grounded verb semantics. This is especially important because the robot's perception of the world is noisy and incomplete, human language can be ambiguous, and the robot may lack the relevant linguistic or world knowledge during the learning process.

To address these limitations, we have developed a new interactive learning approach where robots actively engage with humans to acquire models of grounded verb semantics. Our approach explores the space of interactive question answering between humans and robots during the learning process. In particular, motivated by previous work on robot learning (Cakmak and Thomaz, 2012), we designed a set of questions that are pertinent to verb semantic representations. We further applied reinforcement learning to learn an optimal policy that guides the robot in deciding when to ask what questions. Our empirical results have shown that this interactive learning process leads to more reliable representations of grounded verb semantics, which contribute to significantly better action performance in new situations. When the environment is noisy and uncertain (as in a realistic situation), the models acquired from interactive learning result in a performance gain between 48% and 145% when applied in new situations. Our results further demonstrate that the interaction policy acquired from reinforcement learning leads to the most efficient interaction and the most reliable verb models.

2 Related Work

To enable human-robot communication and collaboration, recent years have seen an increasing amount of works which aim to learn semantics of language that are grounded to agents' perception (Gorniak and Roy, 2007; Tellex et al., 2014; Kim and Mooney, 2012; Matuszek et al., 2012a; Liu et al., 2014; Liu and Chai, 2015; Thomason et al., 2015, 2016; Yang et al., 2016; Gao et al., 2016) and action (Matuszek et al., 2012b; Artzi and Zettlemoyer, 2013; She et al., 2014; Misra et al., 2014, 2015; She and Chai, 2016). Specifically for verb semantics, recent works explored the connection between verbs and action planning (She et al., 2014; Misra et al., 2014, 2015; She and Chai, 2016), for example, by representing grounded verbs semantics as the desired goal state of the physical world that is a result of the corresponding actions. Such representations are learned based on example actions demonstrated by humans. Once acquired, these representations will allow agents to interpret verbs/commands issued by humans in new situations and apply action planning to execute actions. Given its clear advantage in connecting verbs with actions, our work also applies the state-based representation for verb semantics. However, we have developed a new approach which goes beyond learning from demonstrated examples by exploring how rich interaction between humans and agents can be used to acquire models for grounded verb semantics.

This approach was motivated by previous cognitive studies (Bransford et al., 2000) on how people learn as well as recent findings on robot skill learning (Cakmak and Thomaz, 2012). One of the principles for human learning is that "learning is enhanced through socially supported interactions". Studies have shown that social interaction with teachers and peers (e.g., substantive conversation) can enhance student learning experience and improve learning outcomes. In recent work on interactive robot learning of new skills (Cakmak and Thomaz, 2012), researchers identified three types of questions that can be used by a human/robot student to enhance learning outcomes: 1) *demonstration query* (i.e., asking for a full or partial demonstration of the task), 2) *label query* (i.e., asking whether an execution is correct), and 3) *feature query* (i.e., asking for a specific feature or aspect of the task). Inspired by these previous findings, our work explores interactive learning to acquire grounded verb semantics. In particular, we aim to address when to ask what questions during interaction to improve learning.

3 Acquisition of Grounded Verb Semantics

This section gives a brief review on acquisition of grounded verb semantics and illustrates the differences between previous approaches and our approach using interactive learning.

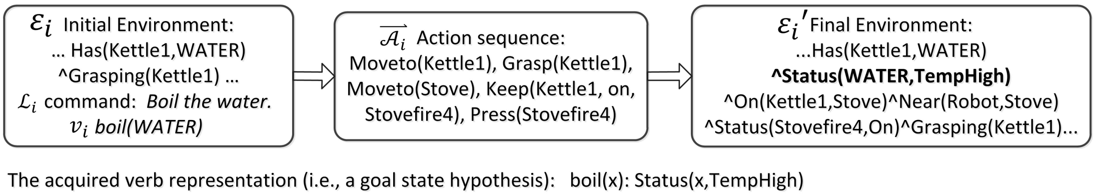
$\mathcal{E}_i$ Initial Environment:
... Has(Kettle1,WATER)
^Grasping(Kettle1) ...
$\mathcal{L}_i$ command: *Boil the water.*
v_i *boil(WATER)*
$\overrightarrow{\mathcal{A}_i}$ Action sequence:
Moveto(Kettle1), Grasp(Kettle1),
Moveto(Stove), Keep(Kettle1, on,
Stovefire4), Press(Stovefire4)
$\mathcal{E}_i{}'$ Final Environment:
...Has(Kettle1,WATER)
^Status(WATER,TempHigh)
^On(Kettle1,Stove)^Near(Robot,Stove)
^Status(Stovefire4,On)^Grasping(Kettle1)...
The acquired verb representation (i.e., a goal state hypothesis): boil(x): Status(x,TempHigh)

Figure 1: An example of acquiring state-based representation for verb semantics based on an initial environment $\mathcal{E}_i$, and a language command $\mathcal{L}_i$, the primitive action sequence $\overrightarrow{\mathcal{A}_i}$ demonstrated by the human, and the final environment $\mathcal{E}_i'$ that results from the execution of $\overrightarrow{\mathcal{A}_i}$ in $\mathcal{E}_i$.

3.1 State-based Representation

As shown in Figure 1, the verb semantics (e.g., *boil(x)*) is represented by the goal state (e.g., $Status(x, TempHigh)$) which is the result of the demonstrated primitive actions. Given the verb phrase *boil the water* (i.e., $\mathcal{L}_i$), the human teaches the robot how to accomplish the corresponding action based on a sequence of primitive actions $\overrightarrow{\mathcal{A}_i}$. By comparing the final environment $\mathcal{E}_i'$ with the initial environment $\mathcal{E}_i$, the robot is able to identify the state change of the environment, which becomes a hypothesis of goal state to represent verb semantics. Compared to procedure-based representations, the state-based representation supports automated planning at the execution time. It is environment-independent and more generalizable. In (She and Chai, 2016), instead of one hypothesis, it maintains a specific-to-general hypothesis space as shown in Figure 2 to capture all goal hypotheses of a particular verb frame. Specifically, it assumes that one verb frame may lead to different outcomes under different environments, where each possible outcome is represented by one node in the hierarchical graph and each node is a conjunction of multiple atomic fluents. [1]

Given a language command (i.e., a verb phrase), a robot will engage in the following processes:

- **Execution.** In this process, the robot will select a hypothesis from the space of hypotheses that is most relevant to the current situation and use the corresponding goal state to plan for actions to execute.

- **Learning.** When the robot fails to select a hypothesis or fails to execute the action, it will ask the human for a demonstration.

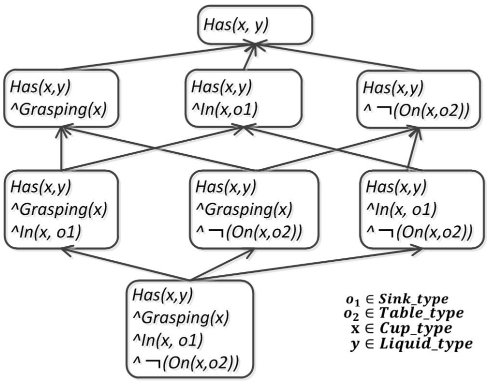
Has(x, y)
Has(x,y) ^Grasping(x)
Has(x,y) ^In(x,o1)
Has(x,y) ^ ¬(On(x,o2))
Has(x,y) ^Grasping(x) ^In(x, o1)
Has(x,y) ^Grasping(x) ^ ¬(On(x,o2))
Has(x,y) ^In(x, o1) ^ ¬(On(x,o2))
Has(x,y) ^Grasping(x) ^In(x, o1) ^ ¬(On(x,o2))
$o_1 \in$ Sink_type
$o_2 \in$ Table_type
$x \in$ Cup_type
$y \in$ Liquid_type

Figure 2: An example hypothesis space for the verb frame $fill(x, y)$.

Based on the demonstrated actions, the robot will learn a new representation (i.e., new nodes) and update the hypothesis space.

3.2 Noisy Environment

Probabilistic Environment:
...^Has(Kettle1,Water) 0.64 ^Grasping(Kettle1) 0.91
^Status(Kettle1,HighTemp) 0.95 ^On(Kettle1,Stove) 0.2
^Near(Robot,Stove) 0.43 ^Status(Stovefire4,On) 0.6 ^...

Figure 3: An example probabilistic sensing result.

Previous works represent the environment $\mathcal{E}_i$ as a conjunction of grounded state fluents. Each fluent consists of a predicate and one or more arguments (i.e., objects in the physical world, or object status), representing one aspect of the perceived environment. An example of a fluent is "$Has(Kettle_1, WATER)$" meaning object $Kettle_1$ has some water inside, where Has is the predicate, and $Kettle_1$ and $WATER$ are arguments. The set of fluents include the status of the robot (e.g., $Grasping(Kettle_1)$), the status of different objects (e.g., $Status(WATER, TempHigh)$), and relations between objects (e.g., $On(Kettle_1, Stove)$). One limitation of the previous works is that the envi-

[1] In this work, we assume the set of atomic fluents representing environment state are given and do not address the question of whether these predicates are adequate to represent a domain.

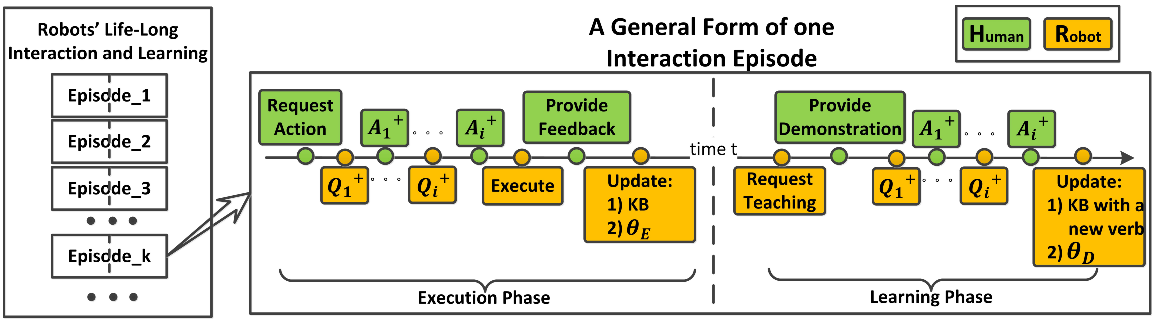

Figure 4: A general framework of robot interactive learning. KB stands for knowledge base, θ_E stands for Interaction Strategy for Execution, and θ_D stands for Interaction Strategy for Learning.

Only demonstration(She and Chai, 2016)	**Interactive learning (this work)**
Execution Example H1: Boil the water. R1: (*select a hypothesis and execute the action accordingly*) H2: (*provide true/false feedback on whether the execution is correct*)	**Execution Example** H1: Boil the water. R1: I see a silver object on the table, is that a pot? H2: Yes. R2: Should I leave the pot on the stove after boil? H3: Yes. R3: Ok. (*select a hypothesis to execute*) H4: Thank you.
Learning Example H3: Boil the water. R3: I don't know how to boil, could you show me an example? H4: (*human shows the robot how to achieve the corresponding action through a sequence of primitive actions*) R4: (*indicate the acceptance and induce a hypothesis to represent verb semantics*)	**Learning Example** H5: Boil the water. R5: I don't know how to boil, could you show me an example? H6: Ok. (*demonstrate the task, at the end put the pot on a heat pad*) R6: So after boiling something, should I put it on a heat pad? H7: No, you don't have to. R7: I see. (*induce goal state hypothesis*)

Table 1: Examples to show differences between learning through demonstrations as in the previous works (She and Chai, 2016) and the proposed learning from interaction.

ronment has a perfect, deterministic representation, as shown in Figure 1. This is clearly not the case in the realistic physical world.

In reality, given limitations of sensor capabilities, the environment representation is often partial, error prone, and full of uncertainties. Figure 3 shows an example of a more realistic representation where each fluent comes with a confidence between 0 and 1 to indicate how likely that particular fluent can be detected in the current environment. Thus, it is unclear whether the previous work is able to handle representations with uncertainties. Our interactive learning approach aims to address these uncertainties through interactive question answering with human partners.

4 Interactive Learning

4.1 Framework of Interactive Learning

Figure 4 shows a general framework for interactive learning of action verbs. It aims to support a life-long learning cycle for robots, where the robot can continuously (1) engage in collaboration and communication with humans based on its existing knowledge; (2) acquire new verbs by learning from humans and experiencing the change of the world (i.e., grounded verb semantics as in this work); and (3) learn how to interact (i.e., update interaction policies). The lifelong learning cycle is composed by a sequence of interactive learning episodes (Episode 1, 2...) where each episode consists of either an execution phase or a learning phase or both.

The execution phase starts with a human request for action (e.g., *boil the water*). According to its interaction policy, the robotic agent may choose to ask one or more questions (i.e., Q_i^+) and wait for human answers (i.e., A_i^+), or select a hypothesis from its existing knowledge base to execute the command (i.e., *Execute*). With the human feedback of the execution, the robot can update its interaction policy and existing knowledge.

In the learning phase, the robot can initiate the learning by requesting a demonstration from the human. After the human performs the task, the robotic agent can either choose to update its knowledge if it feels confident, or it can choose to ask the human one or more questions before updating its knowledge.

4.2 Examples of Interactive Learning

Table 1 illustrates the differences between the previous approach that acquires verb models based solely on demonstrations and our current work that acquires models based on interactive learning. As shown in Table 1, under the *demonstration* setting, humans only provide a demonstration of primitive actions and there's no interactive question answering. In the *interactive learning* setting, the robot can proactively choose to ask questions regarding the uncertainties either about the environment (e.g., R1), the goal (e.g., R2), or the demonstrations (e.g., R6). Our hypothesis is that rich interactions based on question answering will allow the robot to learn more reliable models for grounded verb semantics, especially in a noisy environment.

Then the question is how to manage such interaction: when to ask and what questions to ask to most efficiently acquire reliable models and apply them in execution. Next we describe the application of reinforcement learning to manage interactive question answering for both the execution phase and the learning phase.

4.3 Formulation of Interactive Learning

Markov Decision Process (MDP) and its closely related Reinforcement Learning (RL) have been applied to sequential decision-making problems in dynamic domains with uncertainties, e.g., dialogue/interaction management (Singh et al., 2002; Paek and Pieraccini, 2008; Williams and Zweig, 2016), mapping language commands to actions (Branavan et al., 2009), interactive robot learning (Knox and Stone, 2011), and interactive information retrieval (Li et al., 2017). In this work, we formulate the choice of when to ask what questions during interaction as a sequential decision-making problem and apply reinforcement learning to acquire an optimal policy to manage interaction.

Specifically, each of the execution and learning phases is governed by one policy (i.e., θ_E and θ_D), which is updated by the reinforcement learning algorithm. The use of RL intends to obtain optimal policies that can lead to the highest long-term reward by balancing the cost of interaction (e.g., the length of interaction and difficulties of questions) and the quality of the acquired models. The reinforcement formulation for both the execution phase and the learning phase are described below.

State For the execution phase, each state $s_e \in S_E$ is a five tuple: $s_e = <l, e, KB, Grd, Goal>$. l is a language command, including a verb and multiple noun phrases extracted by the Stanford parser. For example, the command "*Microwave the ramen*" is represented as $l = microwave(ramen)$. The environment e is a probabilistic representation of the currently perceived physical world, consisting of a set of grounded fluents and the confidence of perceiving each fluent (an example is shown in Figure 3). KB stands for the existing knowledge of verb models. Grd accounts for the agent's current belief of object grounding: the probability of each noun in the l being grounded to different objects. $Goal$ represents the agent's belief of different goal state hypotheses of the current command. Within one interaction episode, command l and knowledge KB will stay the same, while e, Grd, and $Goal$ may change accordingly due to interactive question answering and robot actions. In the execution phase, Grd and $Goal$ are initialized with existing knowledge of learned verb models. For the learning phase, a state $s_d \in S_D$ is a four tuple: $s_d = <l, e_{start}, e_{end}, Grd>$. e_{start} and e_{end} stands for the environment before the demonstration and after the demonstration.

Action Motivated by previous studies on how humans ask questions while learning new skills (Cakmak and Thomaz, 2012), the agent's question set includes two categories: yes/no questions and wh- questions. These questions are designed to address ambiguities in noun phrase grounding, uncertain environment sensing, and goal states. They are domain independent in nature. For example, one of the questions is np_grd_ynq(n, o). It is a yes/no question asking whether the noun phrase n refers to an object o (e.g., "*I see a silver object, is that the pot?*"). Other questions are env_pred_ynq(p) (i.e., whether a fluent p is present in the environment; e.g., "*Is the microwave door open?*") and goal_pred_ynq(p) (i.e., whether a predicate p should be part of the goal; "*Should the pot be on a pot stand?*"). Table 2 lists all the actions available in the execution and learning phases. The select_hypo action (i.e., select a goal hypothesis to execute) is only for the execution. Ideally, after asking questions, the agent should be more likely to select a goal hy-

Action Name	Explanation	Question Example	Reward
1. **np_grd_whq**(n)	Ask for the grounding of a np.	*"Which is the cup, can you show me?"*	-6.5[1]
2. **np_grd_ynq**(n, o)	Confirm the grounding of a np.	*"I see a silver object, is that the pot?"*	-1.0 / -2.0
3. **env_pred_ynq**(p)	Confirm a predicate in current environment.	*"Is the microwave door open?"*	-1.0 / -2.0
4. **goal_pred_ynq**(p)	Confirm whether a predicate p should be in the final environment.	*"Is it true the pot should be on the counter?"*	-1.0 / -2.0
5. **select_hypo**(h)	Choose a hypothesis to use as goal and execute.		100 / -2.0
6. **bulk_np_grd_ynq**(n, o)	Confirm the grounding of multiple nps.	*"I think the pot is the red object and milk is in the white box, am I right?"*	-3.0 / -6.0[2]
7. **pred_change_ynq**(p)	Ask whether a predicate p has been changed by the action demonstration.	*"The pot is on a stand after the action, is that correct?"*	-1.0 / -2.0
8. **include_fluent**($\wedge p$)	Include $\wedge p$ into the goal state representation. Update the verb semantic knowledge.		100 / -2.0

Table 2: The action space for reinforcement learning, where n stands for a noun phrase, o a physical object, p a fluent representation of the current state of the world, h a goal hypothesis. Action 1 and 2 are shared by both the execution and learning phases. Action 3, 4, 5 are for the execution phase, and 6, 7, 8 are only used for the learning phase. -1.0/-2.0 are typically used for yes/no questions. When the human answers the question with a "yes", the reward is -1.0, otherwise it's -2.0.

pothesis that best describes the current situation. For the learning phase, the include_fluent($\wedge p$) action forms a goal hypothesis by conjoining a set of fluents ps where each p should have high probability of being part of the goal.

Transition The transition function takes action a in state s, and gives the next state s' according to human feedback. Note that the command l does not change during interaction. But the agent's belief of environment e, object grounding Grd, and goal hypotheses $Goal$ is changed according to the questions and human answers. For example, suppose the agent asks whether noun phrase n refers to the object o, if the human confirms it, the probability of n being grounded to o becomes 1.0, otherwise it will become 0.0.

Algorithm 1: Policy learning. The execution and learning phases share the same learning process, but with different state s, action a spaces, and feature vectors ϕ. The e_{end} is only available to the learning phase.

```
Input      : $e$, $l$ (, $e_{end}$);
             Feature function $\phi$;
             Old policy $\theta$ (i.e., a weight vector)
             Verb Goal States Hypotheses $\mathcal{H}$;
Initialize : state $s$ initialized with $e$, $l$ (, $e_{end}$);
             first action $a \sim P(a|s;\theta)$ with $\epsilon$ greedy
while s is not terminal do
    Take action $a$, receive reward $r$;
    $s' = T(s, a)$;
    Choose $a' \sim P(a'|s';\theta)$ with $\epsilon$ greedy;
    $\delta \leftarrow r + \gamma \cdot \theta^T \cdot \phi(s', a') - \theta^T \cdot \phi(s, a)$;
    $\theta \leftarrow \theta + \delta \cdot \eta \cdot \phi(s, a)$;
end
if s terminates with positive feedback then
    Update $\mathcal{H}$;
end
Output     : Updated $\mathcal{H}$ and $\theta$.
```

Reward Finding a good reward function is a hard problem in reinforcement learning. Our current approach has followed the general practice in the spoken dialogue community (Schatzmann et al., 2006; Fang et al., 2014; Su et al., 2016). The immediate robot questions are assigned small costs to favor shorter and more efficient interaction. Furthermore, motivated by how humans ask questions (Cakmak and Thomaz, 2012), yes/no questions are easier for a human to answer than the open questions (e.g., wh-questions) and thus are given smaller costs. A large positive reward is given at the end of interaction when the task is completed successfully. Detailed reward assignment for different actions are shown in Table 2.

Learning The *SARSA* algorithm with linear function approximation is utilized to update policies θ_E and θ_D (Sutton and Barto, 1998). Specifically, the objective of training is to learn an optimal value function $Q(s, a)$ (i.e., the expected cu-

[1] According to the study in (Cakmak and Thomaz, 2012), the frequency of y/n questions used by humans is about 6.5 times the frequency of open questions (wh question), which motivates our assignment of -6.5 to wh questions.

[2] bulk_np_grd_ynq asks multiple object grounding all at once. This is harder to answer than asking for a single np. Therefore, its cost is assigned three times of the other yes/no questions.

Features shared by both phases
If a is a np_grd_whq(n).
The entropy of candidate groundings of n.
If n has more than 4 grounding candidates.
If a is a np_grd_ynq(n, o).
The probability of n grounded to o.

Additional Features specific for the Execution phase
If a is a select_hypo(h) action.
The probability of hypo h not satisfied in current environment.
Similarity between the ns used by command l and the commands from previous experiences.

Additional Features specific for the Learning phase
If a is a pred_change_ynq(p).
The probability of p been changed by demo.

Table 3: Example features used by the two phases. a stands for action. Other notations are the same as used in Table 2. The"If" features are binary, and the other features are real-valued.

mulative reward of taking action a in a state s). This value function is approximated by a linear function $Q(s, a) = \theta^{\intercal} \cdot \phi(s, a)$, where $\phi(s, a)$ is a feature vector and θ is a weight updated during training. Details of the algorithm is shown in Algorithm 1. During testing, the agent can take an action a that maximizes the Q value at a state s.

Feature Example features used by the two phases are listed in Table 3. These features intend to capture different dimensions of information such as specific types of questions, how well noun phrases are grounded to the environment, uncertainties of the environment, and consistencies between a hypothesis and the current environment.

5 Evaluation

5.1 Experiment Setup

Dataset. To evaluate our approach, we utilized the benchmark made available by (Misra et al., 2015). Individual language commands and corresponding action sequences are extracted similarly as (She and Chai, 2016). This dataset includes common tasks in the kitchen and living room domains, where each data instance comes with a language command (e.g., "*boil the water*", "*throw the beer into the trashcan*") and the corresponding sequence of primitive actions. In total, there are 979 instances, including 75 different verbs and 215 different noun phrases. The length of primitive action sequences range from 1 to 51 with an average of 4.82 (+/-4.8). We divided the dataset into three groups: (1) 200 data instances were used by reinforcement learning to acquire optimal interaction policies; (2) 600 data instances were used by different approaches (i.e., previous approaches and our interactive learning approach) to acquire grounded verb semantics models; and (3) 179 data instances were used as testing data to evaluate the learned verb models. The performance on applying the learned models to execute actions for the testing data is reported.

To learn interaction policies, a simulated human model is created from the dataset (Schatzmann et al., 2006) to continuously interact with the robot learner[3]. This simulated user can answer the robot's different types of questions and make decisions on whether the robot's execution is correct. During policy learning, one data instance can be used multiple times. At each time, the interaction sequence is different due to *exploitation and exploration* in RL in selecting the next action. The RL discount factor γ is set to 0.99, the ϵ in ϵ-greedy is 0.1, and the learning rate is 0.01.

Noisy Environment Representation. The original data provided by (Misra et al., 2015) is based on the assumption that environment sensing is perfect and deterministic. To enable incomplete and noisy environment representation, for each fluent (e.g., $grasping(Cup_3)$, $near(robot_1, Cup_3)$) in the original data, we independently sampled a confidence value to simulate the likelihood that a particular fluent can be detected correctly from the environment. We applied the following four different variations in sampling the confidence values, which correspond to different levels of sensor reliability.

(1) PerfectEnv represents the most reliable sensor. If a fluent is true in the original data, its sampled confidence is 1, and 0 otherwise.

(2) NormStd3 represents a relatively reliable sensor. For each fluent in the original environment, a confidence is sampled according to a normal distribution $\mathcal{N}(1, 0.3^2)$ with an interval [0,1]. This distribution has a large probability of sampling a number larger than 0.5, meaning the corresponding fluent is still more likely to be true.

(3) NormStd5 represents a less reliable sensor. The sampling distribution is $\mathcal{N}(1, 0.5^2)$, which has a larger probability of generating a number smaller than 0.5 compared to *NormStd3*.

[3] In our future work, interacting with real humans will be conducted through Amazon Mechanical Turk. And the policies acquired with a simulated user in this work will be used as initial policies.

(4) UniEnv represents an unreliable sensor. Each number is sampled with a uniform distribution between 0 and 1. This means the sensor works randomly. A fluent has a equal change to be true or false no matter what the true environment is.

Evaluation Metrics. We used the same evaluation metrics as in the previous works (Misra et al., 2015; She and Chai, 2016) to evaluate the performance of applying the learned models to testing instances on action planning.

- ***IED***: Instruction Editing Distance. This is a number between 0 and 1 measuring the similarity between the predicted action sequence and the ground-truth action sequence. ***IED*** equals 1 if the two sequences are exactly the same.

- ***SJI***: State Jaccard Index. This is a number between 0 and 1 measuring the similarity between the predicted and the ground-truth state changes. ***SJI*** equals 1 if action planning leads to exactly the same state change as in the ground-truth.

Configurations. To understand the role of interactive learning in model acquisition and action planning, we first compared the interactive learning approach with the previous leading approach (presented as *She16*). To further evaluate the interaction policies acquired by reinforcement learning, we also compared the learned policy (i.e., ***RLPolicy***) with the following two baseline policies:

- ***RandomPolicy*** which randomly selects questions to ask during interaction.

- ***ManualPolicy*** which continuously asks for yes/no confirmations (i.e., object grounding questions ($GroundQ$), environment questions ($EnvQ$), goal prediction questions ($GoalQ$)) until there's no more questions before making a decision on model acquisition or action execution.

5.2 Results

5.2.1 The Effect of Interactive Learning

	She16		*RL policy*		% improvement	
	IED	*SJI*	*IED*	*SJI*	*IED*	*SJI*
PerfectEnv	0.430	0.426	0.453	0.468	5.3%*	9.9%*
NormStd3	0.284	0.273	0.420	0.431	47.9%*	57.9%*
NormStd5	0.172	0.168	0.392	0.411	127.9%*	144.6%*
UniEnv	0.168	0.163	0.332	0.347	97.6%*	112.9%*

Table 4: Performance comparison between *She16* and our interactive learning based on environment representations with different levels of noise. All the improvements (marked *) are statistically significant ($p < 0.01$).

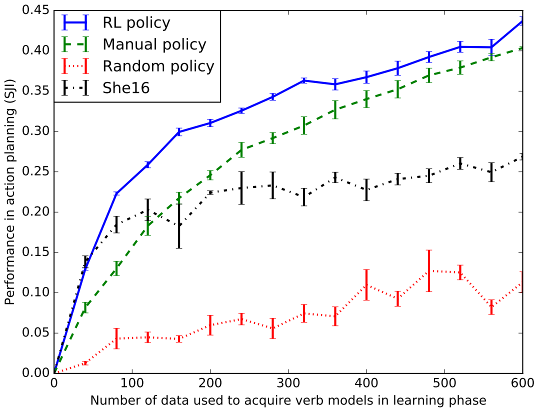

Figure 5: Performance (SJI) comparison by applying models acquired based on different interaction policies to the testing data.

Table 4 shows the performance comparison on the testing data between the previous approach *She16* and our interactive learning approach based on environment representations with different levels of noise. The verb models acquired by interactive learning perform better consistently across all four environment conditions. When the environment becomes noisy (i.e., *NormStd3*, *NormStd5*, and *UniEnv*), the performance of *She16* that only relies on demonstrations decreases significantly. While the interactive learning improves the performance under the perfect environment condition, its effect in noisy environment is more remarkable. It leads to a significant performance gain between 48% and 145%. These results validate our hypothesis that interactive question answering can help to alleviate the problem of uncertainties in environment representation and goal prediction.

Figure 5 shows the performance of the various learned models on the testing data, based on a varying number of training instances and different interaction policies. The interactive learning guided by the policy acquired from RL outperforms the previous approach *She16*. The RL policy slightly outperforms interactive learning using manually defined policy (i.e., *ManualPolicy*). However, as shown in the next section, the *Man-*

	Average number of questions							Performance	
	Learning Phase			Execution Phase					
	GroundQ	*EnvQ*	*TotalQ*	*GroundQ*	*EnvQ*	*GoalQ*	*TotalQ*	*IED*	*SJI*
RLPolicy	2.130* +/-0.231	2.615* +/-0.317	4.746* +/-0.307	0.383* +/-0.137	0.650* +/-0.366	2.626 +/-0.331	3.665* +/-0.469	0.420 +/-0.015	0.430* +/-0.018
ManualPolicy	2.495 +/-0.025	5.338 +/-0.008	7.833 +/-0.025	1.236 +/-0.002	3.202 +/-0.012	2.353 +/-0.023	6.792 +/-0.025	0.406 +/-0.002	0.404 +/-0.004
RandomPolicy	0.545 +/-0.016	0.368 +/-0.033	0.913 +/-0.040	0.678 +/-0.055	0.081 +/-0.030	0.151 +/-0.024	0.909 +/-0.018	0.114 +/-0.025	0.113 +/-0.029

Table 5: Comparison between different policies including the average number (and standard deviation) of different types of questions asked during the execution phase and the learning phase respectively, and the performance on action planning for the testing data. The results are based on the noisy environment sampled by *NormStd3*. $*$ indicates statistically significant difference ($p < 0.05$) comparing *RLPolicy* with *ManualPolicy*.

ualPolicy results in much longer interaction (i.e., more questions) than the RL acquired policy.

5.2.2 Comparison of Interaction Policies

Table 5 compares the performance of different interaction policies. It shows the average number of questions asked under different policies. It is not surprising the *RandomPolicy* has the worst performance. For the *ManualPolicy*, its performance is similar to the *RLPolicy*. However, the average interaction length of *ManualPolicy* is 6.792, which is much longer than the *RLPolicy* (which is 3.127). These results further demonstrate that the policy learned from RL enables efficient interactions and the acquisition of more reliable verb models.

6 Conclusion

Robots live in a noisy environment. Due to the limitations in their external sensors, their representations of the shared environment can be error prone and full of uncertainties. As shown in previous work (Mourão et al., 2012), learning action models from the noisy and incomplete observation of the world is extremely challenging. The same problem applies to the acquisition of verb semantics that are grounded to the perceived world. To address this problem, this paper presents an interactive learning approach which aims to handle uncertainties of the environment as well as incompleteness and conflicts in state representation by asking human partners intelligent questions. The interaction strategies are learned through reinforcement learning. Our empirical results have shown a significant improvement in model acquisition and action prediction. When applying the learned models in new situations, the models acquired through interactive learning leads to over 140% performance gain in noisy environment.

The current investigation also has several limitations. As in previous works, we assume the world can be described by a closed set of predicates. This causes significant simplification for the physical world. One of the important questions to address in the future is how to learn new predicates through interaction with humans. Another limitation is that the current utility function is learned based on a set of pre-identified features. Future work can explore deep neural network to alleviate feature engineering.

As cognitive robots start to enter our daily lives, data-driven approaches to learning may not be possible in new situations. Human partners who work side-by-side with these cognitive robots are great resources that the robots can directly learn from. Recent years have seen an increasing amount of work on task learning from human partners (Saunders et al., 2006; Chernova and Veloso, 2008; Cantrell et al., 2012; Mohan et al., 2013; Asada et al., 2009; Mohseni-Kabir et al., 2015; Nejati et al., 2006; Liu et al., 2016). Our future work will incorporate interactive learning of verb semantics with task learning to enable autonomy that can learn by communicating with humans.

Acknowledgments

This work was supported by the National Science Foundation (IIS-1208390 and IIS-1617682) and the DARPA SIMPLEX program under a subcontract from UCLA (N66001-15-C-4035). The authors would like to thank Dipendra K. Misra and colleagues for providing the evaluation data, and the anonymous reviewers for valuable comments.

References

Yoav Artzi and Luke Zettlemoyer. 2013. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics* Volume1(1):49–62.

Minoru Asada, Koh Hosoda, Yasuo Kuniyoshi, Hiroshi Ishiguro, Toshio Inui, Yuichiro Yoshikawa, Masaki Ogino, and Chisato Yoshida. 2009. Cognitive developmental robotics: A survey. *IEEE Transactions on Autonomous Mental Development* 1(1):12–34.

S. R. K. Branavan, Harr Chen, Luke S. Zettlemoyer, and Regina Barzilay. 2009. Reinforcement learning for mapping instructions to actions. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '09, pages 82–90.

John D. Bransford, Ann L. Brown, and Rodney R. Cocking. 2000. *How People Learn: Brain, Mind, Experience, and School: Expanded Edition*. National Academy Press., Washington, DC.

Maya Cakmak and Andrea L. Thomaz. 2012. Designing robot learners that ask good questions. In *Proceedings of the 7th Annual ACM/IEEE International Conference on Human-Robot Interaction*. ACM, New York, NY, USA, HRI '12, pages 17–24.

R. Cantrell, K. Talamadupula, P. Schermerhorn, J. Benton, S. Kambhampati, and M. Scheutz. 2012. Tell me when and why to do it! run-time planner model updates via natural language instruction. In *Proceedings of the 7th Annual ACM/IEEE International Conference on Human-Robot Interaction*. ACM, New York, NY, USA, HRI '12, pages 471–478.

Sonia Chernova and Manuela Veloso. 2008. Teaching multi-robot coordination using demonstration of communication and state sharing. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3*. International Foundation for Autonomous Agents and Multiagent Systems, pages 1183–1186.

Rui Fang, Malcolm Doering, and Joyce Y. Chai. 2014. Collaborative models for referring expression generation in situated dialogue. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*. AAAI Press, AAAI'14, pages 1544–1550.

Qiaozi Gao, Malcolm Doering, Shaohua Yang, and Joyce Y. Chai. 2016. Physical causality of action verbs in grounded language understanding. In *ACL (1)*. The Association for Computer Linguistics.

P. Gorniak and D. Roy. 2007. Situated language understanding as filtering perceived affordances. In *Cognitive Science*, volume 31(2), pages 197–231.

Joohyun Kim and Raymond J. Mooney. 2012. Unsupervised pcfg induction for grounded language learning with highly ambiguous supervision. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Natural Language Learning (EMNLP-CoNLL '12)*. Jeju Island, Korea, pages 433–444.

W. Bradley Knox and Peter Stone. 2011. Understanding human teaching modalities in reinforcement learning environments: A preliminary report. In *IJCAI 2011 Workshop on Agents Learning Interactively from Human Teachers (ALIHT)*.

Jiwei Li, Alexander H. Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. 2017. Learning through dialogue interactions. In *ICLR*.

Changsong Liu and Joyce Y. Chai. 2015. Learning to mediate perceptual differences in situated human-robot dialogue. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI Press, AAAI'15, pages 2288–2294.

Changsong Liu, Lanbo She, Rui Fang, and Joyce Y. Chai. 2014. Probabilistic labeling for efficient referential grounding based on collaborative discourse. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 13–18.

Changsong Liu, Shaohua Yang, Sari Saba-Sadiya, Nishant Shukla, Yunzhong He, Song-chun Zhu, and Joyce Y. Chai. 2016. Jointly learning grounded task structures from language instruction and visual demonstration. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 1482–1492.

Cynthia Matuszek, Nicholas Fitzgerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. 2012a. A joint model of language and perception for grounded attribute learning. In John Langford and Joelle Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*. ACM, New York, NY, USA, pages 1671–1678.

Cynthia Matuszek, Evan Herbst, Luke S. Zettlemoyer, and Dieter Fox. 2012b. Learning to parse natural language commands to a robot control system. In Jaydev P. Desai, Gregory Dudek, Oussama Khatib, and Vijay Kumar, editors, *ISER*. Springer, volume 88 of *Springer Tracts in Advanced Robotics*, pages 403–415.

Dipendra K Misra, Jaeyong Sung, Kevin Lee, and Ashutosh Saxena. 2014. Tell me dave: Context-sensitive grounding of natural language to manipulation instructions. *Proceedings of Robotics: Science and Systems (RSS), Berkeley, USA* .

Dipendra Kumar Misra, Kejia Tao, Percy Liang, and Ashutosh Saxena. 2015. Environment-driven lexicon induction for high-level instructions. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the*

7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, Beijing, China, pages 992–1002.

Shiwali Mohan, James Kirk, and John Laird. 2013. A computational model for situated task learning with interactive instruction. In *Proceedings of ICCM 2013 - 12th International Conference on Cognitive Modeling*.

Anahita Mohseni-Kabir, Charles Rich, Sonia Chernova, Candace L. Sidner, and Daniel Miller. 2015. Interactive hierarchical task learning from a single demonstration. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*. ACM, HRI '15, pages 205–212.

Kira Mourão, Luke S. Zettlemoyer, Ronald P. A. Petrick, and Mark Steedman. 2012. Learning STRIPS operators from noisy and incomplete observations. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*. Catalina Island, CA, USA, pages 614–623.

Negin Nejati, Pat Langley, and Tolga Konik. 2006. Learning hierarchical task networks by observation. In *Proceedings of the 23rd international conference on Machine learning*. ACM, pages 665–672.

Tim Paek and Roberto Pieraccini. 2008. Automating spoken dialogue management design using machine learning: An industry perspective. *Speech Communication* 50(8-9):716–729.

Joe Saunders, Chrystopher L Nehaniv, and Kerstin Dautenhahn. 2006. Teaching robots by moulding behavior and scaffolding the environment. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*. ACM, pages 118–125.

Jost Schatzmann, Karl Weilhammer, Matt Stuttle, and Steve Young. 2006. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *Knowl. Eng. Rev.* 21(2):97–126.

Lanbo She and Joyce Y. Chai. 2016. Incremental acquisition of verb hypothesis space towards physical world interaction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Lanbo She, Shaohua Yang, Yu Cheng, Yunyi Jia, Joyce Y. Chai, and Ning Xi. 2014. Back to the blocks world: Learning new actions through situated human-robot dialogue. In *Proceedings of the SIGDIAL 2014 Conference, The 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 18-20 June 2014, Philadelphia, PA, USA*. pages 89–97.

Satinder Singh, Diane Litman, Michael Kearns, and Marilyn Walker. 2002. Optimizing dialogue management with reinforcement learning: Experiments with the njfun system. *Journal of Artificial Intelligence Research* 16:105–133.

Pei-Hao Su, Milica Gasic, Nikola Mrkšić, Lina M. Rojas Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. On-line active reward learning for policy optimisation in spoken dialogue systems. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 2431–2441.

Richard S. Sutton and Andrew G. Barto. 1998. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition.

Stefanie Tellex, Pratiksha Thaker, Joshua Joseph, and Nicholas Roy. 2014. Learning perceptually grounded word meanings from unaligned parallel data. *Machine Learning* 94(2):151–167.

Jesse Thomason, Jivko Sinapov, Maxwell Svetlik, Peter Stone, and Raymond J. Mooney. 2016. Learning multi-modal grounded linguistic semantics by playing "i spy". In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI-16)*. New York City, pages 3477–3483.

Jesse Thomason, Shiqi Zhang, Raymond Mooney, and Peter Stone. 2015. Learning to interpret natural language commands through human-robot dialog. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*. pages 1923–1929.

Jason D Williams and Geoffrey Zweig. 2016. End-to-end lstm-based dialog control optimized with supervised and reinforcement learning. *arXiv preprint arXiv:1606.01269* .

Shaohua Yang, Qiaozi Gao, Changsong Liu, Caiming Xiong, Song-Chun Zhu, and Joyce Y. Chai. 2016. Grounded semantic role labeling. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*. pages 149–159.