# Tutelage and Socially Guided Robot Learning

Andrea Lockerd, Cynthia Breazeal
MIT Media Lab
20 Ames St., Cambridge MA, 02139
alockerd, cynthiab @media.mit.edu

***Abstract*— We view the problem of machine learning as a collaboration between the human and the machine. Inspired by human-style tutelage, we situate the learning problem within a dialog in which social interaction structures the learning experience, providing instruction, directing attention, and controlling the complexity of the task. We present a learning mechanism, implemented on a humanoid robot, to demonstrate that a collaborative dialog framework allows a robot to efficiently learn a task from a human, generalize this ability to a new task configuration, and show commitment to the overall goal of the learned task. We also compare this approach to traditional machine learning approaches.**

## I. Introduction

If robots are to serve useful long term roles in our lives (doing household chores, assisting at work, caring for the sick and elderly), the ability for naive users to teach them new skills easily will be key to their success. Additionally, they must learn new skills in a goal-oriented fashion that is robust to changes in the environment. In this paper we consider the ways in which teaching a robot can more closely resemble the natural ease of human tutelage, a paradigm shift in the machine learning problem.

Learning by human tutelage leverages from structure provided through interpersonal interaction. It is a collaboration between the teacher and the learner. Teachers direct a learner's attention, structure their experiences, support their learning attempts, and regulate the complexity of information. The learner contributes by revealing their internal state to help guide the teaching process. Tutelage is a fundamentally collaborative process, enabling a learner to acquire new concepts and skills from few examples.

We frame the machine learning problem as a collaborative dialog between the human teacher and the robot learner to take advantage of the wealth of structure and information this interaction provides. Here we describe our latest work exploring how social guidance interacts with traditional inference algorithms (such as Bayesian hypothesis testing) in an interactive learning scenario. Our implementation allows a human to teach a robot a new task using dialog and gesture. The robot communicates its current understanding through demonstration and expressive social cues. Over a few trials, the robot learns a generalized task goal and can apply this concept in a new configuration of the task state.

## II. Background

Situated learning is a field of study that looks at the social world of a child and how it contributes to their development. One key concept is ”scaffolding”, where an adult organizes a new skill into manageable steps and provides support such that a child can achieve something they would not be able to accomplish independently [16].

In a situated learning interaction, a good instructor maintains an accurate mental model of the learner's understanding and structures the learning task appropriately with timely feedback and guidance. The learner contributes to the process by expressing their internal state via communicative acts (e.g., expressing understanding, confusion, attention, etc.). This reciprocal and tightly coupled interaction enables the learner to leverage from instruction to build the appropriate representations and associations.

This situated learning process stands in contrast to typical scenarios of machine learning which are often not interactive nor intuitive for the human partner. Our motivation is the belief that the human can provide more than labeled examples or a reinforcement signal. Active learning [6] more closely resembles the interaction we envision. Typically this approach has the machine identify the 'most interesting' examples with unsupervised learning, then the human is queried for labels of just these seminal examples. This metaphor could extend even further, such that the machine not only explicitly queries the human teacher but can take advantage of social cues situated in the tutelage process.

## III. System Architecture

Our research platform is Leonardo (“Leo”), a humanoid robot with 65 degrees of freedom that has been specifically designed for social interaction (Fig. 1). Leo currently relies on gestures and facial expressions for communication.

The robot has both speech and visual inputs. The vision system parses objects from the visual scene such as humans and the robot's toys (e.g., buttons that it can press). These perceptions are sent to the cognitive system along with object attributes (e.g., color, location). The vision system also recognizes pointing gestures and uses spatial reasoning to associate these gestures with their object referent. The speech understanding system, based on the NRL Nautilus project [12], is a Lisp parser with a ViaVoice front end. Nautilus receives phrases from ViaVoice and parses them into information that is sent to the cognitive system.

The cognitive system extends the `C5M` architecture, see [3]. It receives data continuously from the vision and speech understanding systems and integrates these into coherent beliefs about objects in the world, gestures, and speech. Visual perceptions are merged based on location and kept in one structure (e.g. information about a particular toy is merged with the corresponding features of

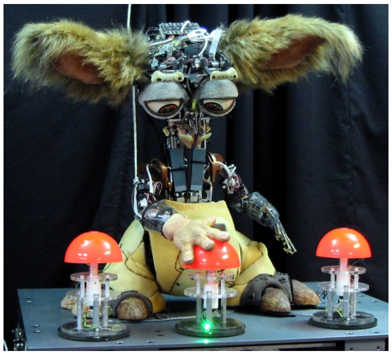

Fig. 1. Leo and his workspace with three button toys.

location, color and state to form one coherent belief). The cognitive system can also manipulate beliefs internally to add information (e.g., adding a label when a person names an object). Additionally, we have developed capabilities to support socially guided learning.

## IV. Task Learning and Execution

During teaching, the human interactively instructs the robot, building a new task from its set of known actions and tasks. Each task example yields a number of potential task representation hypotheses, letting Leo build a flexible goal-oriented hierarchical task model. Executing tasks and incorporating feedback narrows the hypothesis space, converging to their best representation. The following sections describe the implementation of tasks, goals, hypotheses, and the hypothesis testing process.

### A. Task Representation

Tasks and their constituent actions are variations of the `C5M` *action-tuple* structure; the notion of *goals* is embodied in the until-condition. The learning module continually pays attention to actions being performed, encoding inferred goals upon completion of any action, sub-task, or task. The resulting hierarchical representation, has an overall task goal as well as goals for each component [5].

Given the natural ambiguity of learning, the task representation should be flexible. Our task representation contains a set of hypothesis task representations from which the current best hypothesis is used for execution. A task hypothesis has executables, a goal representation, and a record of the number of examples that have been consistent with this particular task representation. A task executable is one of three types: a primitive action, a sub-task, or an abstract action (a result of generalization discussed below).

### B. Goal Representation

Many studies show that humans interpret actions based on goals and intentions rather than motion trajectories [1], [17]. This is crucial in a collaborative setting, where goals are the common ground for communication and interaction [5]. Thus our task representation is goal-oriented, supporting a realistic groundwork for intentional understanding–i.e. performing the task to accomplish the *overall intent*, rather than mechanically performing the constituent actions.

When an action completes the robot infers the goal and saves this action to the current task. Similarly when a task completes, the robot determines the ’overarching goal’ of the learned task. There are two goal types: (a) `state-change` goals represent a world change, and (b) `just-do-it` goals need to be executed regardless of their impact. Leo compares the world state before and after and action or task and if there is no change, it is considered a `just-do-it` goal. Otherwise it is a `state-change` goal, but there is ambiguity around what exactly about the end state was the goal (the change to an object, a class of objects, the whole world, etc.). Our approach uses hypothesis testing coupled with human interaction to disambiguate the task goal over a few examples.

Goals are object-based (objects are co-located sets of perceived features). A task goal is a set of *beliefs* about what must hold true in order to consider this task achieved. A `just-do-it` goal has a single goal belief. For a `state-change` goal, a *goal belief* is made for each object that incurred a change during the action or task. A belief has *criteria* features and *expectation* features. Criteria features are those that held constant over the task and expectation features are the aspects of the object that changed. For example, upon finishing Task-X the world changed such that a particular toy A is unchanged and toy B changed from green to red. The goal for Task-X is a `state-change` goal with one goal belief containing criteria features [type=toy, location=xyz, label=B, etc] and an expectation feature [color=red]. This allows for a straightforward evaluation of task goals: for each goal belief, find an object with the criteria features and check that the expectation features hold true.

### C. Expanding Task Hypotheses

When the human indicates that the task is done, the task manager adds this task to the collection of known tasks. For a `state-change` goal, the system expands hypotheses about the goal state. If the actions performed are similar–i.e. all of the task actions are the same primitive type with different objects of attention, then this primitive action is the *generalized task action* (GTA). Next the system looks at each of goal belief (each changed object) and forms a Common Goal Belief (CGB) containing all the criteria and expectation features in common. A number of task hypotheses are made where the executable is the GTA and the goal is one of the combinations of goal features from the CGB. For example, if the CGB has 4 features, one hypothesis will be the GTA and a goal belief with all 4 features (the most specific hypothesis). Another hypothesis will be the GTA and a goal belief with 3 of the 4 features, etc. This expansion yields a hypothesis space of all representations consistent with the current task example.

The current best task representation is chosen through a Bayesian likelihood method. The likelihood of each of the hypotheses is calculated according to Bayes rule ($P(h|D) \propto P(D|h)P(h)$). The probability of a hypothesis,

TABLE I

SOCIAL CUES FOR SCAFFOLDING

| Context | Leo's Expression | Intention |
|---|---|---|
| Human points to object | Looks at Object | Shows Object of Attention |
| Executing an Action | Looks at Object | Shows Object of Attention |
| Human: "Let's learn task X" | Subtle Head Nod | Confirms start of task X |
| Human: "Task X is done" | Subtle Head Nod | Confirms end of task X |
| Any speech | Perks ears | Conveys that Leo is listening |
| Unconfident task execution | Glances to human frequently | Conveys uncertainty |
| Completion of demonstration | Perks ears, lean forward | Soliciting feedback from teacher |
| Human: "Can you...?" | Perform or Nod/Shake | Communicates task knowledge |
| Human: "Do task X" | Performs X | Demonstrating hypothesis for X |
| Task done; Human: "Not quite..." | Subtle nod | Confirms feedback, expects refinement |
| Task done; Human: "Good!" | Nods head | Confirms task hypothesis |
| Turn-taking Dialog | Eye contact | Making/breaking eye contact frames turns |

*h*, given data, *D*, is proportional to the probability of the data given the hypothesis scaled by the prior likelihood. In our case, the data, *D*, is the set of all examples seen for this task. $P(D|h)$ is the percentage of the examples in which the state change seen in the example is consistent with the goal representation in *h*. For priors, $P(h)$, our algorithm prefers a more specific hypothesis over a more general one. For example, when a task is first learned, every hypothesis is equally represented in the data and the algorithm chooses the most specific representation for the next execution.

### D. Task Execution

When Leo is asked to do a known task, and the goal is incomplete, the task manager starts the execution by expanding the task's executables onto a *focus stack* (as in [7]). Execution then works through the actions on the stack popping them as they are done or, for a sub-task, testing its goal before pushing its constituent actions onto the stack.

The task hypothesis used for execution has a likelihood (between 0 and 1) relative to the other hypotheses available. If this confidence is low ($< .5$), Leo expresses tentativeness (frequently looking between the instructor and an action's object of attention). Upon finishing the task, Leo leans forward with his ears perked waiting for feedback. The teacher can give positive verbal feedback and Leo considers the task complete. Or, if Leo has not yet achieved the goal they can give negative verbal feedback and Leo will expect the teacher to lead him through the completion of the task. A new example is created through this refinement stage, similar to the original learning process.

Once task execution completes (whether it required refinement or not), a task hypothesis is created along with the expanded hypothesis space for this example (this is identical to the expansion process described in the initial learning phase). For each hypothesis, if it exists in the task hypothesis space already the coverage count is incremented, otherwise it is added to the space. Again, with the Bayesian likelihood method, the best hypothesis is chosen for the next execution of this task.

### E. Expression and Gestural Communication

The robot cooperates in the teaching/learning collaboration by maintaining a mutual belief with the teacher about the task state, expressing confusion, understanding, attention, etc. A number of expressive skills contribute to Leo's effectiveness in learning through collaborative dialog (Table I). Eye gaze establishes joint attention, reassuring the teacher that the robot is paying attention to the right thing. Subtle nods acknowledge task stages, confirming a mutual understanding of moving on to the next stage.

In a realistic robot interaction, the speech recognition system is not perfect and will occasionally not be able to parse the human's utterance. To naturally overcome this roadblock Leo perks his ears as soon as the human begins speaking to indicate that he is paying attention. If unable to parse this speech, Leo will gesture (leaning forward with hand to ear) to indicate that speech recognition failed and the human needs to repeat their last phrase.

The robot uses expressions to indicate to the human tutor when he is ready to learn something new, and demonstration of taught actions provides immediate feedback about task comprehension. When performing a recently taught task, ear and body position as well as eye gaze are used to solicit feedback from the human when uncertainty is high. By frequently looking back at the human during the performance, Leo signals to the teacher that confidence is low, soliciting feedback and further examples.

## V. EVALUATION

We have tested Leonardo's socially guided learning abilities on several tasks with simple dialogs and manipulation skills. In the experimental scenario, there are three colored buttons in front of Leo. The buttons can be pressed ON or OFF, switching an LED on or off. Occasionally, a button does not light up when pressed and this is considered a failed attempt. We have successfully taught tasks of both simple and complex hierarchies, and tasks with both `state-change` and `just-do-it` goals. Including, turning a set of buttons ON then OFF, and turning a button ON as a single action or as a sub-task

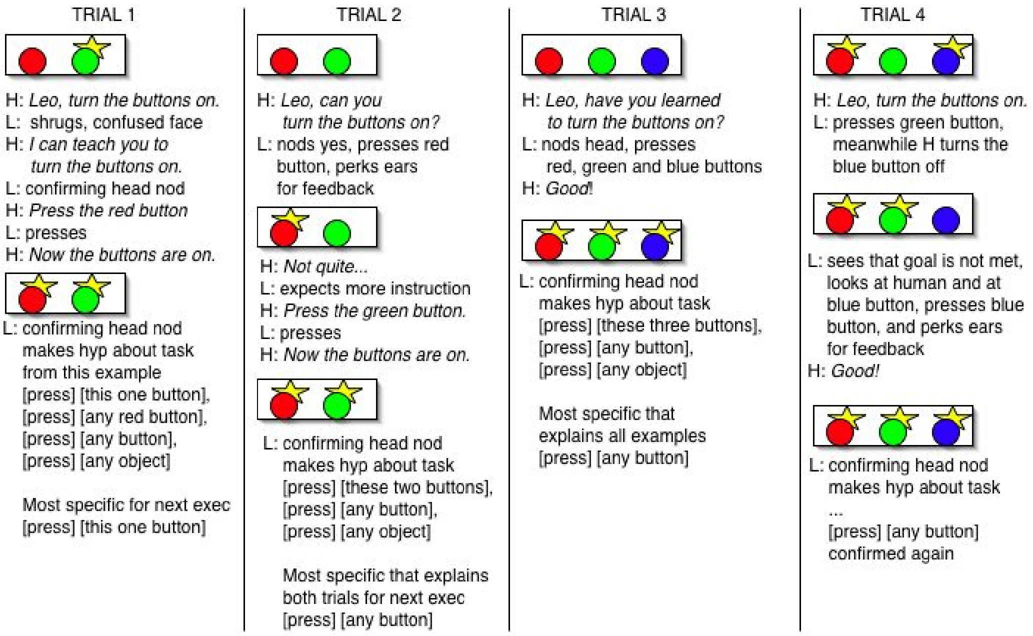
TRIAL 1
H: Leo, turn the buttons on.
L: shrugs, confused face
H: I can teach you to turn the buttons on.
L: confirming head nod
H: Press the red button
L: presses
H: Now the buttons are on.
L: confirming head nod makes hyp about task from this example
[press] [this one button],
[press] [any red button],
[press] [any button],
[press] [any object]
Most specific for next exec
[press] [this one button]
TRIAL 2
H: Leo, can you turn the buttons on?
L: nods yes, presses red button, perks ears for feedback
H: Not quite...
L: expects more instruction
H: Press the green button.
L: presses
H: Now the buttons are on.
L: confirming head nod makes hyp about task
[press] [these two buttons],
[press] [any button],
[press] [any object]
Most specific that explains both trials for next exec
[press] [any button]
TRIAL 3
H: Leo, have you learned to turn the buttons on?
L: nods head, presses red, green and blue buttons
H: Good!
L: confirming head nod makes hyp about task
[press] [these three buttons],
[press] [any button],
[press] [any object]
Most specific that explains all examples
[press] [any button]
TRIAL 4
H: Leo, turn the buttons on.
L: presses green button, meanwhile H turns the blue button off
L: sees that goal is not met, looks at human and at blue button, presses blue button, and perks ears for feedback
H: Good!
L: confirming head nod makes hyp about task
...
[press] [any button]
confirmed again

Fig. 2. An interaction in which a human (H) teaches Leo (L) to 'Turn the buttons ON'. From left to right the button colors are: Red, Green, Blue.

of turning all of the buttons ON, to name a few. The robot recalled tasks which were learned as sub-tasks of larger activities, demonstrating understanding of nested action. Understanding when to perform an action and how long to persist based on initial success, showed correct association of `state-change` goals and `just-do-it` goals.

Here we present one task example in detail, learning to 'turn all the buttons ON', and discuss the socially guided learning approach compared to a typical machine learning approach. Reinforcement learning is a reasonable machine learning approach for this experimental scenario. Thus we present a typical reinforcement learning approach, Q-Learning [14], for learning this task.

One might argue that this task is a toy example, but we believe it is an important starting point. Learning a task made up of known actions is quite generalizable to more complex behavior. Consider Robonaut, envisioned as a robotic astronaut's assistant [2]. Many complex behaviors that a human might want to teach Robonaut will in fact decompose into a hierarchical representation of simpler known actions and tasks. Moreover, even our simple example demonstrates important differences between our approach and a standard reinforcement learning approach.

Technically the two algorithms are quite different, and the Q-learning was only run in simulation. However, it is useful to consider the differences in the socially guided learning approach and a standard reinforcement learning approach. Particularly we will focus on the differences in what is actually learned and the differences in the interaction that the human has with the learning algorithm.

### A. Socially Guided Learning

Fig. 2 shows a socially guided learning session where a human taught Leo to 'turn the buttons ON' (beginning with only a few primitive actions: press, look, point, etc.). In Trial 1, the green button was already on, so the human only had to ask Leo to press the red button. This led to a variety of hypotheses about the actual task representation, but the most specific is chosen for the next execution. In Trial 2, the human purposely structures the task to resolve an ambiguity in the previous example. After this example, the most specific hypothesis that explains both examples is [press][any button], the correct representation. He exhibits the correct behavior in Trial 3, and shows commitment to the overarching goal in Trial 4. This example is in a low dimensional feature space and little data is needed to resolve the ambiguities, but even here the advantage of the tutelage paradigm is shown. Through a turn-taking interaction with a human partner, the robot quickly acquires the representative examples needed to generalize to the correct task representation. The robot's expressions and demonstration of learned abilities helps the teacher understand what the robot knows and what ambiguities remain.

### B. Q-Learning

For Q-Learning, the state space consists of all ON/OFF configurations of the buttons, and the actions are the different button presses. The transition model between states is assumed to be deterministic, but a more realistic transition model would be probabilistic (allowing for button failure). The reward function assigns a (+1) for reaching the 'all buttons ON' goal state and (0) for every other state. From a particular state, the action policy chooses the action with the maximum value; thus, the goal of the Q-Learning algorithm is to learn the value of the various state-action pairs (the Q-values). We simulated three different configurations of this approach. In each of the configurations, a trial starts in a random non-terminal state. The algorithm proceeds,

TABLE II
Q-LEARNING RESULTS - EACH AVERAGED OVER 25 RUNS

| Button config | avg Trials | (min, max) | avg Acts | (min, max) |
|---|---|---|---|---|
| 3 (init .5) | 12 | (5, 35) | 45 | (34, 80) |
| 2 (init .5) | 3 | (1, 5) | 9 | (5, 12) |
| 3 (init w/ 2) | 9 | (3, 16) | 31 | (17, 44) |

choosing the maximum valued action and then updating the Q-value based on the reward signal received, and the trial completes when the goal state is reached. Trials were continually run (restarting in a random state) until the algorithm converged to the optimal action policy (i.e. for any starting state of the buttons, the least number of presses is used to reach the goal state).

The first configuration learns the action policy for the case where only two buttons are visible (similar to Trials 1 and 2 in Fig. 2). The size of the state space is 4, and there are 2 actions available in every state. All of the 8 Q-values were initialized to .5.

The second configuration learns the action policy for the case where all three buttons are visible. The size of the state space is 8, and there are 3 actions available in every state. Again, all 24 Q-values were initialized to .5.

The third configuration learns an action policy for the three buttons case, but instead of initializing all 24 Q-values equally, the Q-values related to actions 0 and 1 are seeded with their values from the two-button Q-value table. For example, if the state-action pair $\{01, 0\}$ has the value .25 in the two-button case, the state-action pairs $\{001, 0\}$ and $\{101, 0\}$ are both initialized with the value .25 in the three-button case. All of the state-action pairs related to action 2 (which has no information in the two-button case) are initialized equally to the maximum value from the two-button case (to promote exploration).

Table II has the results of the various Q-learning configurations, from training each configuration 25 times. 'Trials' indicates the number of trails before the algorithm showed optimal performance, and 'Actions' is the total number of actions required in these trials.

Note that the use of the two-button policy values did not provide much speed up in the learning of the three button policy; on average the algorithm still needed another 9 trials beyond the 3 it took to learn the two button policy, for a total of 12. In terms of interaction, with the Q-Learning approach a person training the robot to 'turn the buttons ON' would wait for the robot to explore the space and then only give feedback when the goal state is reached.

## VI. DISCUSSION

In this work we explore a different view of the machine learning problem, viewing the teaching/learning problem as a collaboration between the robot and the human partner, and using human social skills to constrain and guide the learning process. More than a good human-robot interaction technique, the ability to utilize and leverage from human social skills can positively impact the underlying learning mechanism. In this section we highlight the differences between our approach and a standard machine learning approach, followed by discussion of related work.

### A. *Goal-Oriented Task Representation*

The most significant difference between our approach and others is what is actually being learned. Leo is learning the goal of the new task. In other approaches, like Q Learning, the goal is encoded in the reinforcement signal a priori and the machine learns the best way to achieve it. In our approach, once Leo knows the goal he can generate a way to achieve it without having to explore. Moreover, a goal-based representation provides a generalizable representation of the task, leading to a realistic generalization to additional task configurations. Thus, the robot is flexible in how it performs tasks: the configuration of the task can change (Fig. 2, Trial 3), or a partially completed state can be presented and the robot knows how to complete it (Fig. 2, Trial 4). A Reinforcement Learning policy will not generalize since environmental changes (adding a button) alter the state-action space and require more learning. Even seeding the learning process with information from the prior configuration still needs quite a bit of learning. With the goal-oriented approach, the robot learns not only the actions to perform, but to achieve a desired result and demonstrate commitment to the goal.

### B. *Transparent Learning*

Continually communicating using demonstrations, expressive gestures, and eye gaze, Leo maintains a mutual belief with the human teacher about the learning state which helps guide the teaching process. The teacher can interactively structure the task, providing more seminal examples as learning progresses. Thus, the human is actively narrowing down the hypothesis space with the robot. This transparency leads to more relevant and timely instruction and thus to more efficient learning. Conversely, in more ”opaque” learning systems it is difficult for the teacher to understand the learning process or see why the system is doing the wrong thing. Q-Learning, for example, explores the state-action space, not taking advantage of specific guidance from the human partner. In teaching the three buttons ON task, instead of guiding the robot the human waits through about 45 press actions and only interacts with the robot the 12 times it reaches the goal.

### C. *Just-in-Time Correction*

The turn-taking dialog framework lets the teacher know right away what problems or issues remain unclear, enabling just-in-time error correction with refinement to a failed attempts. Through gesture and eye gaze, the robot lets the teacher know when the current task representation has a low confidence, naturally soliciting feedback and further examples. In contrast to typical reinforcement learning, where feedback propagates after a complete attempt and potentially has credit assignment problems; in the socially guided approach the human's feedback leads to refinement that efficiently corrects errors as they arise.

### D. Related Work

Discourse analysis and collaborative dialog theory have been used in plan recognition [10] and tutorial systems [13]. However, we use collaborative dialog to frame the robot learning from a human rather than tutoring a human.

Other methods of robots learning from people include learning by demonstration or observation [8], [15]. Generally such approaches use demonstration to seed the search space, optimizing a pre-defined goal function. Our aim is that the robot learn *new* goals from human interaction.

In Instruction-Based Learning [9], a human uses natural language to instruct a robot in a navigation task. The instruction is prior to execution, requiring that the human mentally represent the whole task prior to its execution. We believe the situated aspect of tutelage is key for both the learner and the teacher. In learning by doing, our robot makes its own assessment of the world to build a task representation. Socially guided learning also simplifies the teacher's task; leading the robot through the task allows for adjustments and corrections at appropriate times.

Nicolescu and Matarić [11] have inspiration similar to ours with a tutelage paradigm in which a robot learns a sequential task from human demonstration (e.g., learning a path through an environment of objects). The robot follows the human through the task, and the human uses short verbal commands to point out information and frame the demonstration. When the robot performs the task, the human can interrupt to correct the task model. Our work goes further in modeling learning as a collaborative process, with a more tightly coupled turn-taking framework. We use social expression to form a natural dialog between the teacher and the learner to transparently express the robot's internal state and solicit feedback and instruction.

The issue of transparency in a learning system is also addressed by [4]. While our inspiration is human tutelage and theirs is animal training, their interactive dog character is an example of a system that learns from guided exploration and is rewarding to teach.

## VII. Conclusion

We aim to enable robots to learn new tasks from natural human instruction with ordinary people (not experts in robotics or machine learning). People are often benevolent teachers motivated to help the robot learn. Structuring and guidance through interpersonal interaction will be natural for them— it makes sense to build teachable robots that take advantage of this. Inspired by human tutelage, we have presented our first demonstration that a collaborative socially guided learning approach can improve both the human-robot interaction and the machine learning process. Within a collaborative dialog, the robot continually communicates its internal state to the human partner, cooperating to maintain mutual beliefs with the human teacher such that they can appropriately guide the learning process. In future work we hope to show that natural social interaction can apply to other machine learning algorithms as a new approach to the machine learning process

## Acknowledgment

This work results from ongoing efforts of the MIT Media Lab Robotic Life Group and our collaborators. Leo's speech understanding abilities are a collaboration with Alan Schultz and his group at the Naval Research Lab. Stan Winston Studio provided the physical robot. Leonardo's architecture uses the C5M code base of the MIT Media Lab Synthetic Characters Group. Geoff Beatty and Ryan Kavanaugh provided the virtual model and animations. This work is funded in part by a DARPA MARS grant and in part by the Digital Life and Things that Think consortia.

## References

[1] D.A. Baldwin and J.A. Baird. Discerning intentions in dynamic human action. *Trends in Cognitive Sciences*, 5(4):171–178, 2001.

[2] William Bluethmann, Robert Ambrose, Myron Diftler, Scott Askew, Eric Huber, Michael Goza, Fredrik Rehnmark, Chris Lovchik, and Darby Magruder. Robonaut: A robot designed to work with humans in space. *Auton. Robots*, 14(2-3):179–197, 2003.

[3] B. Blumberg, R. Burke, D. Isla, M. Downie, and Y. Ivanov. CreatureSmarts: The art and architecture of a virtual brain. In *Proceedings of the Game Developers Conference*, pages 147–166, 2001.

[4] B. Blumberg, M. Downie, Y. Ivanov, M. Berlin, M.P. Johnson, and B. Tomlinson. Integrated learning for interactive synthetic characters. In *Proceedings of the ACM SIGGRAPH*, 2002.

[5] C. Breazeal, G. Hoffman, and A. Lockerd. Teaching and working with robots as collaboration. In *Proceedings of the AAMAS*, 2004.

[6] D. Cohn, Z. Ghahramani, and M. Jordan. Active learning with statistical models. In G. Tesauro, D. Touretzky, and J. Alspector, editors, *Advances in Neural Information Processing*, volume 7. Morgan Kaufmann, 1995.

[7] B. J. Grosz and C. L. Sidner. Plans for discourse. In P. R. Cohen, J. Morgan, and M. E. Pollack, editors, *Intentions in communication*, chapter 20, pages 417–444. MIT Press, Cambridge, MA, 1990.

[8] Y. Kuniyoshi, M. Inaba, and H. Inoue. Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation*, 10:799–822, 1994.

[9] Stanislao Lauria, Guido Bugmann, Theocharis Kyriacou, and Ewan Klein. Mobile robot programming using natural language. *Robotics and Autonomous Systems*, 38(3-4):171–181, 2002.

[10] Diane J. Litman and James F. Allen. Discourse processing and commonsense plans. In P. R. Cohen, J. Morgan, and M. E. Pollack, editors, *Intentions in communication*, chapter 17, pages 417–444. MIT Press, Cambridge, MA, 1990.

[11] Monica N. Nicolescu and Maja J. Matarić. Natural methods for robot task learning: Instructive demonstrations, generalization and practice. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Melbourne, Australia, July 2003.

[12] Dennis Perzanowski, Alan C. Schultz, William Adams, Elaine Marsh, and Magda Bugajska. Building a multimodal human-robot interface. *IEEE Intelligent Systems*, 16(1):16–21, 2001.

[13] C. Rich, C. Sidner, and N. Lesh. Collagen: Applying collaborative discourse theory to human-computer interaction. *AI Magazine*, pages 15–25, 2001.

[14] R. S. Sutton and A. G. Barto. In *Reinforcement learning: An introduction*. MIT Press, Cambridge, MA, 1998.

[15] R. Voyles and P. Khosla. A multi-agent system for programming robotic agents by human demonstration. In *Proceedings of AI and Manufacturing Research Planning Workshop*, 1998.

[16] L. S. Vygotsky. In M. Cole, editor, *Mind in society: the development of higher psychological processes*. Harvard University Press, Cambridge, MA, 1978.

[17] A. L. Woodward, J. A. Sommerville, and J. J. Guajardo. How infants make sense of intentional actions. In B. Malle, L. Moses, and D. Baldwin, editors, *Intentions and Intentionality: Foundations of Social Cognition*, chapter 7, pages 149–169. MIT Press, Cambridge, MA, 2001.