Asymmetric Interpretations of Positive and Negative Human Feedback for a Social Learning Agent

Andrea L. Thomaz, Cynthia Breazeal
MIT Media Lab, Cambridge, MA 02139, USA,
alockerd@media.mit.edu, cynthiab@media.mit.edu

***Abstract*— The ability for people to interact with robots and teach them new skills will be crucial to the successful application of robots in everyday human environments. In order to design agents that learn efficiently and effectively from their instruction, it is important to understand how people, that are not experts in Machine Learning or robotics, will try to teach social robots. In prior work we have shown that human trainers use positive and negative feedback differentially when interacting with a Reinforcement Learning agent. In this paper we present experiments and implementations on two platforms, a robotic and a computer game platform, that explore the multiple communicative intents of positive and negative feedback from a human partner, in particular that negative feedback is both about the past and about intentions for future action.**

I. Introduction

Social learning will be crucial to the successful application of robots in everyday human environments. It will be virtually impossible to give these machines all of the knowledge and skills a priori that they will need to serve useful long term roles in our dynamic world. The ability for naïve users, not experts, to guide them easily will be key to their success. While recognizing the success of current Machine Learning techniques over the years, these techniques have not been designed for learning from non-expert users and are generally not suited for it 'out of the box'.

A cornerstone of our research is that machines designed to interact with people to learn new things should use behaviors and conventions that are socially relevant to the humans with which they interact. They should more fully be able to participate in the teaching and learning partnership, a two-way collaboration. Moreover, the ability to utilize and leverage these social skills is more than a good interface for people, it can positively impact the underlying learning mechanisms to let the system succeed in a real-time interactive learning session.

In this paper we address a particular aspect of a social learning collaboration: the asymmetric meaning of positive and negative feedback from a human teacher. The intuition is that positive feedback tells a learner undeniably, "what you just did was good." However, negative feedback tells the learner both that the last action was bad, and that the current state is bad and future actions should correct that. Thus, negative feedback is about the past and about future intentions for action.

We present an experiment with a Reinforcement Learning agent that experimentally illustrates the biased nature of positive and negative feedback from a human partner. We then present two implementations that represent two interpretations of negative feedback from a human partner. Both assume that negative feedback from a human partner is feedback about the action or task performed and at the same time communicates something about what should follow.

II. Research Platforms

In this paper we describe experiments and implementations on two research platforms. One is a 65 degree-of-freedom anthropomorphic social robot, Leonardo, and the other is a virtual robot, Sophie, in a computer game platform.

A. Leonardo: Task Learning via Collaborative Social Dialog

Leonardo ("Leo") is a 65 degree of freedom athropomorphic robot specifically designed for social interaction, using a range of facial and body pose expressions (see Figure 1). Leo has both speech and vision perceptual inputs, and relies on gestures and facial expression for social communication. The cognitive system extends the `C5M` architecture, described in [2]. From a stream of symbols from vision and speech understanding systems, it creates coherent beliefs about objects in the world. It also has higher-level cognitive capabilities for representing and learning goal-oriented tasks, as well as expression and gesture capabilities to support a natural collaborative dialog with a human teacher.

In task learning, Leo has a basis set of actions that it can perform on objects in its environment, and a human partner can interactively instruct the robot, building a new task model from its set of known actions and tasks. Thus, task learning is embedded within a tightly-coupled collaborative dialog. Each trial yields a number of potential hypotheses about the task and goal representation. Executing tasks and incorporating feedback narrows the hypothesis space, converging on the best representation. This paper will focus on how the learning mechanism incorporates feedback from the human partner within the social dialog. For further details on Leo's task learning ability see [4] and [18].

B. Sophie's Kitchen: Interactive Reinforcement Learning

Sophie's Kitchen is a Java-based simulation platform. The scenario is a kitchen environment (Fig. 2), where the agent,

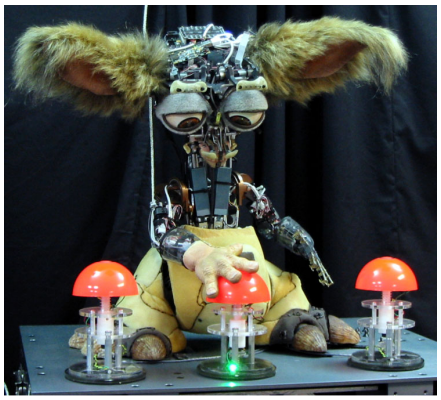

Fig. 1. Leo and his workspace with three button toys.

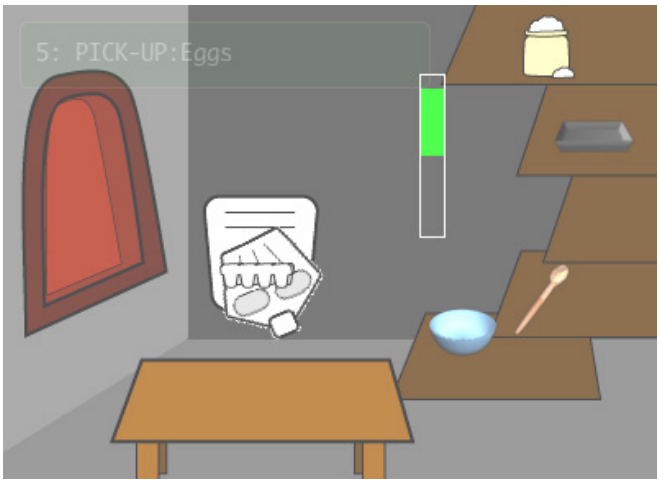

Fig. 2. *Sophie's Kitchen*: There are three locations (oven, table, shelf), and five baking objects (flour, eggs, pan, spoon, and bowl). The virtual robot, Sophie, learns to bake a cake via Q-Learning, and a human partner playing the game can contribute by issuing feedback with the mouse, creating a red or green bar for positive/negative feedback. The green bar seen here is the interactive human feedback.

Sophie, learns to bake a cake using Q-learning [27]. The kitchen creates a sufficiently complex domain with on the order of 10,000 states, and 2-7 actions available from each state.

In the initial state, the agent faces all the objects on the `Shelf`. A successful task completion includes putting flour and eggs in the bowl, stirring the ingredients using the spoon, putting the batter into the tray, and putting the tray in the oven. The agent can `GO left` or `right` to change location; she can `PICK-UP` any object in her current location; she can `PUT-DOWN` any object in her possession; and she can `USE` any object in her possession on any object in her current location.

A central feature of *Sophie's Kitchen* is the interactive reward interface. Using the mouse, a human trainer can—at any point—award a scalar reward signal, $r \in [-1, 1]$. The user receives visual feedback enabling them to tune the reward before sending it, and choosing and sending the reward does not halt the progress of the agent, which runs asynchronously to the reward interface. *Sophie's Kitchen* is

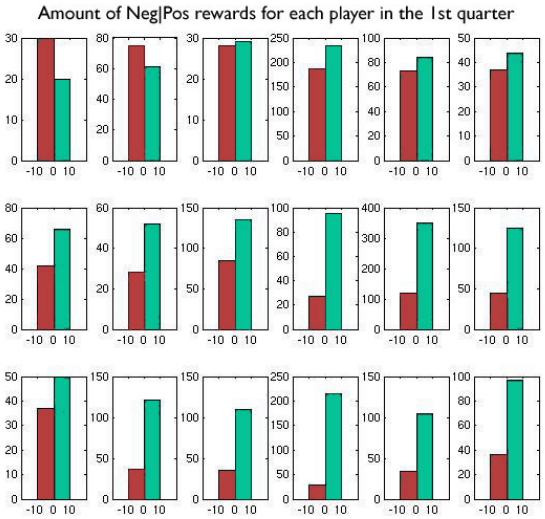

Fig. 3. Histograms of rewards for each individual in the first quarter of their session. The left column is negative rewards and the right is positive rewards. Most people even in the first quarter of training have a much higher bar on the right.

a platform for experimenting with *Interactive* Reinforcement Learning; for more details on the design and implementation of this platform see [26].

III. Asymmetry of Human Feedback

We did an investigative study with the *Sophie's Kitchen* platform [26], in which 18 volunteers played the computer game. Their goal was to get the virtual robot, Sophie, to learn how to bake a cake. Participants were told they could not tell Sophie what actions to do, nor could they do any actions directly. They were only able to send messages with the mouse to help Sophie learn the task.

In this experiment, people have direct control of the reward signal of the agent's Q-Learning algorithm. The purpose of the experiment was to understand, when given a single reward channel, how do people use it to teach the agent? This experiment found that, perhaps not surprisingly, people have multiple communicative intentions beyond the simple feedback that the Q-Learning algorithm expects.

For many people, a large majority of rewards given were positive. The mean percentage of positive rewards for players was 69.8%, (standard deviation: 9). This was thought at first to be due to the agent improving and exhibiting more correct behavior over time (soliciting more positive rewards); however, the data from the first quarter of training shows that well before the agent is behaving correctly, the majority of participants still show a positive bias. Fig. 3 shows reward histograms for each participant's first quarter of training; the number of negative rewards on the left and positive rewards on the right. Most participants have a much larger bar on the right.

Clearly, people have asymmetric intentions they are communicating with their positive and negative feedback messages. In following section we present a specific interpreta-

tion of this asymmetry and implementations on the Sophie and Leonardo platforms that show how this can impact agents that learn from human partners.

IV. Feedback about the Past and the Future

Positive and negative feedback, in the context of human social learning, have asymmetric meanings. Positive feedback says to a learner, "what you did was good". Negative feedback, on the other hand, has a couple of meanings it conveys: 1) the last action was bad, and/or 2) the current state is bad and future actions should correct that. Thus, negative feedback is about both the past and about future intentions for action.

The following implementations present two interpretations of negative feedback. Both assume that negative feedback from a human partner is feedback about the action or task performed and at the same time communicates something about what should follow. In the first example, Leonoardo assumes that negative feedback will lead to refinement of the performed task example. In the second example, Sophie assumes that a negatively reinforced action should be reversed if possible.

A. Negative Feedback Leading to Refined Instruction

Just-in-time error correction is an aspect of Leonardo's Task Learning implementation that represents an asymmetric interpretation of positive and negative feedback from a human partner. During the collaborative learning dialog, when Leonardo demonstrates a learned task, positive feedback reinforces a task hypothesis, but negative feedback leads directly to refinement of the hypothesis.

This approach is drawn from speech act theory, in particular the concept that speakers intend their larger purposes to be inferred from their utterances [6]. In the case of Leonardo, by gesturing in a way to solicit feedback after a demonstration the robot is asking: "Was that the right thing to do?" It is assumed that if the human answers this question they will infer the larger purpose of the joint activity, which implies some commitment to a more than a yes/no response. If the human were to simply answer "no," this does not represent a commitment to the larger joint activity of helping Leo correctly learn the task.

1) Task Execution and Refinement: In Leonardo's task learning, the human stands opposite Leo in his workspace (pictured in Figure 1), and uses speech and gestures to help Leonardo build representations of new tasks/skills based on an initial set of primitive known actions. The *Task Learning Module* maintains the collection of known tasks and arbitrates between task learning and execution. When an unknown task is requested, Leo starts the learning process. The human walks the robot through the components of the task, requesting it to perform the necessary steps to reach the goal, building a new task from its set of known actions and tasks.

While in learning mode, the Task Learning Module continually pays attention to what actions the robot is being asked to perform, encoding the inferred goals with these actions. In order to encode the goal state of a performed action or task, Leo compares the world state before and after its execution. When the human indicates that the task is done, it is added to the Task Learning Module's collection of known tasks.

When Leo is asked to do a known task, and the goal is incomplete, Leo uses the current best task hypothesis for execution, which has a likelihood (between 0 and 1) relative to the other hypotheses available. If this confidence is low, 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 (e.g., "Good," "Good job," "Well done," ...) and Leo considers the task complete and the executed hypothesis gains value (i.e., the number of seen examples consistent with this hypothesis is incremented).

After completing the demonstration, if Leo has not yet achieved the goal the human can give negative verbal feedback (e.g., "No," "Not quite," ...) and Leo will expect the teacher to lead him through the completion of the task. A new example is created through this refinement stage. Leo makes a representation of the change over the task and the actions that were necessary to complete it (the actions he did himself, plus the actions the human requested during refinement).

2) 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 failed attempts. Through gesture and eye gaze, the robot lets the teacher know when the current task representation has a low confidence, soliciting feedback and further examples.

A similar goal concept learning could be achieved with a supervised learning approach that uses batches of positive and negative examples to learn the concept. However, this does not take advantage of the tightly coupled interactive component of learning from a human teacher. Leonardo's on-line interactive learning session lets the human partner provide examples incrementally. They see through demonstration the current state of Leo's goal concept, and are able to interactively make additions to a negative example to change it into a positive example of the goal concept.

B. Negative Feedback as Direction for Exploration

The *Sophie's Kitchen* platform is used to explore another aspect of reward asymmetry. In this approach, negative feedback communicates information both to the learning mechanism updating the policy (in the same way as positive rewards), and also to the action selection mechanism. This implementation shows significant improvements in multiple aspects of learning performance with a human partner, allowing the agent to have a more efficient and robust exploration strategy.

Positive reward for a performed action gives a clear message to the agent - that the probability of performing that action in that state should be increased. A symmetric approach would have the opposite reaction to negative reward

Algorithm 1 Interactive Q-Learning with the addition of the UNDO behavior

while learning **do**
  if (reward last cycle $< -.25$) and (can UNDO last action, a_{last}) **then**
    $a = undo(a_{last})$
  else
    $a =$ random select weighted by $Q[s, a]$ values
  end if
  execute a, and transition to s'
  (small delay to allow for human reward)
  sense reward, r
  update policy:

$$Q[s,a] \leftarrow Q[s,a] + \alpha(r + \gamma(max_{a'} Q[s',a']) - Q[s,a])$$

end while

- decreasing the probability of performing that action in that state. While learning will occur in the symmetric case (the success of several renditions of Reinforcement Learning algorithms are proof), this neglects part of the information communicated by a negative reward.

In addition to communicating that the decision to make that action was wrong, negative feedback communicates that this line of behavior or reasoning is bad. Thus a reaction that more closely resembles intuition about natural learning, is to adopt the goal of being back in the state that one was in before the negative feedback occurred. In many cases, of course not all, actions performed by an agent in the world are reversible. Thus upon negative feedback that agent should first update its value function to incorporate this feedback from the world, but this negative feedback should also communicate with the action selection mechanism that the next action should be a reversal if possible.

We implemented this behavior on the *Sophie's Kitchen* platform and an experiment shows that this behavior leads to more robust learning, keeping the agent in the positive areas of the world, approaching the boundaries but avoiding the negative spaces. This is particularly important for applications in robotic agents acting in the real world with physical hardware that may not withstand much negative interaction with the world. This behavior also generates more efficient learning, reducing both the total time necessary and the number of trials that end in failure.

1) Modification for Sophie's UNDO Response: In the initial experiment, the agent used a basic Q-Learning algorithm. To experiment with asymmetric responses to negative feedback, this baseline algorithm was modified to respond to negative feedback with an UNDO behavior (a natural correlate or opposite action) when possible. Thus a negative reward affects the value function in the normal fashion, but also alters the subsequent action selection if possible. The proper UNDO behavior is represented within each primitive action and is accessed with an $UNDO$ function: the action GO [direction] returns GO [-direction], the action PICK-UP [object] returns PUT-DOWN [object], the action PUT-DOWN [object] returns PICK-UP [object], the USE actions are not reversible. Algorithm 1 shows how this is implemented as a simple modification to the standard Q-Learning algorithm.

2) Evaluation: Experimental data was collected from 97 non-expert human participants by deploying the *Sophie's Kitchen* game on the World Wide Web. They were asked to help the agent learn to bake the cake by sending feedback messages as she makes attempts. When they felt Sophie could bake the cake herself they pressed a button to test the agent and obtained their score (based on how many actions it took for the agent to bake the cake on her own).

The *Sophie's Kitchen* platform offers a measurable comparison between two conditions of the learning algorithm. In the baseline case the algorithm handles both positive and negative feedback in a standard way, feedback is incorporated into the value function. In the undo case the algorithm uses feedback to update the value function but then also uses negative feedback in the action selection stage as an indication that the best action to perform next is the reverse of the negatively reinforced action (Alg. 1). Statistically significant differences were found between the baseline and UNDO conditions on a number of learning performance metrics (summarized in table I).

- Training Failure Reduction: The UNDO behavior helps the agent avoid failure. The total number of failures during the learning phase was significantly less in the UNDO case, 37% less, $t(96) = -3.77$, $p < .001$. This is particularly interesting for robotic agents that need to learn in the real world. For these agents, learning from failure may not be a viable option; thus, utilizing a negative feedback signal to learn the task while avoiding disaster states is necessary.
  The UNDO case also had significantly less failures before the first goal was reached, 40% less, $t(96) = -3.70$, $p < .001$. Related to the overall number of failures being less, there were also less failures before the first success. This is especially important when the agent is learning with a human partner. The human partner will have a limited patience and will need to see progress quickly in order to remain engaged in the task. Thus, the UNDO behavior seems to be a good technique for reaching the first success faster.
- Training Time Efficiency: There was a nearly significant effect for the number of actions required to learn the task, 12% less, $t(96) = -1.32$, $p = .09$, with the UNDO condition requiring less steps (the high degree of variance in the number of steps needed to learn the task leads to the higher p value). Thus, the algorithm that uses the UNDO behavior is able to learn the task in less time (fewer total actions taken).
- Exploration Efficiency Another indication of the efficiency of the UNDO case compared to the baseline is in the state space needed to learn the task. The number of unique states visited is significantly less in the UNDO case, 13% less, $t(96) = -2.26$, $p = .01$. This indicates

TABLE I

1-TAILED T-TEST: SIGNIFICANT DIFFERENCES WERE FOUND BETWEEN THE BASELINE AND UNDO CONDITIONS, IN TRAINING SESSIONS WITH NEARLY 100 NON-EXPERT HUMAN SUBJECTS PLAYING THE *Sophie's Kitchen* GAME ONLINE.

Measure	Mean baseline	Mean UNDO	chg	t(96)	p
# states	48.3	42	13%	-2.26	=.01
# F	6.94	4.37	37%	-3.76	<.001
# F before G	6.4	3.87	40%	-3.7	<.001
# actions to G	208.86	164.93	21%	-2.25	=.01
# actions	255.68	224.2	12%	-1.32	=.095

that when the algorithm interprets negative feedback as a directive for reversing the previous action, or returning to the previous state, the resulting behavior is more efficient in its use of the state space to learn the desired task. Thus, the learning agent stays 'on the right track' in its exploration.

V. DISCUSSION

In Reinforcement Learning it is usual to represent the distinction between appetitive and aversive evaluative feedback using just the sign of a scalar reward signal, where positive means good; negative means bad. Since RL algorithms are based on the objective of maximizing the sum of rewards over time, this makes sense: positive feedback increases the sum; negative feedback decreases it. But we see, from our initial experiment with *Sophie's Kitchen* (Sec. III), that when a human partner is asked to train an RL agent, they do not use the reward channel in symmetric ways.

Furthermore, it is clear that biological systems do not have symmetric responses to positive and negative feedback. Evidence from neuroscience shows that the human brain processes appetitive and aversive rewards differently. Positive and negative feedback stimulate physically different locations in the brain: the left side of the amygdala responds to positive reinforcement, while the right responds to negative reinforcement [28]. Additionally, there is evidence for an 'error processing' mechanism where the anterior cingulate cortex generates signals correlated with error detection (independent of task goal or modality) [9]. This evidence alone does not tell us how or why to include the asymmetry of feedback in our computational learning model, but it does inspire us to search for computational grounds for such inclusion with the goal of developing more efficient and robust learning algorithms. This paper has presented two such computational implementations for treating appetitive and aversive feedback differently.

In the first example, the Leonardo robot assumes that a task demonstration followed by negative feedback will lead to refinement of that example. This is a departure from the normal formulation of supervised learning, where the agent receives a bag of positive and negative examples (or perhaps collects these online over time). In this case the agent has seen only positive examples, and expands hypothesis goal representations. Upon executing a task based on one of these hypotheses, and getting negative feedback, Leo expects the human partner to lead him through refining the example. This lets the agent at once label the hypothesis as bad and at the same time add another positive example to its set. Thus refining the hypothesis space with the human partner.

In *Sophie's Kitchen* on the other hand, the agent takes a different view of negative feedback. It assumes that negative feedback should lead to reversing an action if possible. In the kitchen world, many of the actions are reversible, such that the previous state can be easily achieved. In other more complex system, one could imagine the agent may need to make a plan of action to achieve the previous state, or learn which actions are or are not reversible. In the modified Sophie agent, if negative reinforcement is received and the last action performed is reversible the agent choses this as the next action rather than using its normal action selection mechanism. In experiments with human trainers, this version of the Sophie agent exhibits significantly better learning behavior. The size of the state space visited is much smaller, there are significantly fewer failures, and fewer actions are needed to learn the task.

Finally it is interesting to address the simultaneous use of the two implementations shown in this paper. At first glance they may seem incompatible, however, the approaches represent two strategies that represent different levels of dependence on the human partner's guidance. In the Leo example the assumption is that *more* needs to be done from the current state and the human partner is guiding the additional steps. On the other hand, the Sophie example shows the utility of reverting to the previous state and trying again. Thus, waiting for refinement is a response to negative feedback that places a lot of dependence on the human partner, while 'UNDO' or 'do over' is a response to negative feedback that places more emphasis on self-exploration. In the end, a learning agent is likely to need the ability to use both strategies, the ideal social learning agent should be able to both learn on its own but take full advantage of the human partner if they are present and offering support.

VI. RELATED WORK IN INTERACTIVE LEARNING

For years researchers working on robotic and software agents have been inspired by the idea of efficiently transferring knowledge about tasks or skills from a human to a machine. There are several related works that explicitly incorporate human input to a Machine Learning process. This input or feedback is utilized in various ways across these works.

Many prior works have dealt with the scenario where a machine learns by observing human behavior: personalization agents [15], [10], programming by example [16], robot learning by observation [14], demonstration [1], and imitation [23], [5]. Several examples of human interaction with Reinforcement Learning are inspired by animal training techniques like clicker training and shaping [3], [12], [22]. Related to this, a common technique for incorporating human input to a reinforcement learner lets the human directly control the reward signal to the agent [11], [8], [25]. Other works

let a human supervise a RL agent by directly controlling the training action sequence: specifying teaching sequences [17], or directly controlling the actions actions of a robot agent [24]. Loosening the burden on the human teacher, some methods let the human supervise an RL agent by occasionally biasing action selection rather than directly controlling all of the agent's actions [7], [13], [19].

Recently there have been related robot learning systems that use human interaction to frame and refine the learning process. Nicolescu and Mataric have a robot system that learns a navigation task by following a human demonstrator [20]. The teacher uses simple voice cues to frame the learning ("here," "take," "drop," "stop"), and the robot generalizes a task model over multiple trials with the human. In their teaching dialog the teacher's use of "bad" and "stop" are an example of using negative feedback for refinement. Rybski et al. have a similar approach, with additional emphasis on the refining stage of task learning [21]. They have the robot ask for information it deems necessary rather than rely on feedback from the human teacher.

The contribution of this paper to interactive Machine Learning is the illustration of asymmetry in people's usage of positive and negative feedback to a learning agent, and the development of strategies, based on this asymmetry, that allow a learning agent to better interpret the communicative intents of human teachers. Additionally, prior works generally have not evaluated their interactive learning techniques with everyday human subjects, which is a fundamental aspect of our work.

VII. Conclusion

In order to design social robots that learn efficiently and effectively from people in everyday human environments, it is important to understand how people, that are not experts in Machine Learning or robotics, will try to teach them. Prior work has shown that human trainers use positive and negative feedback differentially when interacting with an RL agent. In this paper we have presented experiments and implementations, on a robotic and a computer game platform, that explore the asymmetric intents of positive and negative feedback from a human partner. On the Leonardo robot, negative feedback leads directly to refinement, a collaborative dialog approach. On the *Sophie's Kitchen* platform, negative feedback leads to action reversal, which results in a significantly faster and more efficient learning behavior.

References

[1] Christopher G. Atkeson and Stefan Schaal. Robot learning from demonstration. In *Proc. 14th International Conference on Machine Learning*, pages 12–20. Morgan Kaufmann, 1997.

[2] 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.

[3] 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.

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

[5] C. Breazeal and B. Scassellati. Robots that imitate humans. *Trends in Cognitive Science*, 6(11), 2002.

[6] H. H. Clark. *Using Language*. Cambridge University Press, Cambridge, 1996.

[7] J. Clouse and P. Utgoff. A teaching method for reinforcement learning. In *Proc. of the Nineth International Conf. on Machine Learning (ICML)*, pages 92–101, 1992.

[8] R. Evans. Varieties of learning. In S. Rabin, editor, *AI Game Programming Wisdom*, pages 567–578. Charles River Media, Hingham, MA, 2002.

[9] C. Holroyd and M. Coles. The neural basis of human error processing: Reinforcement learning, dopamine, and the error-related negativity. *Psychological Review*, 109(4):679–709, 2002.

[10] E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse. The lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 256–265, Madison, WI, July 1998.

[11] C. Isbell, C. Shelton, M. Kearns, S. Singh, and P. Stone. Cobot: A social reinforcement learning agent. *5th Intern. Conf. on Autonomous Agents*, 2001.

[12] F. Kaplan, P-Y. Oudeyer, E. Kubinyi, and A. Miklosi. Robotic clicker training. *Robotics and Autonomous Systems*, 38(3-4):197–206, 2002.

[13] G. Kuhlmann, P. Stone, R. J. Mooney, and J. W. Shavlik. Guiding a reinforcement learner with natural language advice: Initial results in robocup soccer. In *Proceedings of the AAAI-2004 Workshop on Supervisory Control of Learning and Adaptive Systems*, San Jose, CA, July 2004.

[14] 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.

[15] Y. Lashkari, M. Metral, and P. Maes. Collaborative Interface Agents. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, volume 1. AAAI Press, Seattle, WA, 1994.

[16] H. Lieberman, editor. *Your Wish is My Command: Programming by Example*. Morgan Kaufmann, San Francisco, 2001.

[17] L. J. Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8:293–321, 1992.

[18] A. Lockerd and C. Breazeal. Tutelage and socially guided robot learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[19] R. Maclin, J. Shavlik, L. Torrey, T. Walker, and E. Wild. Giving advice about preferred actions to reinforcement learners via knowledge-based kernel regression. In *Proceedings of the The Twentieth National Conference on Artificial Intelligence (AAAI)*, Pittsburgh, PA, July 2005.

[20] M. N. Nicolescu and M. J. Matarić. Natural methods for robot task learning: Instructive demonstrations, generalization and practice. In *Proceedings of the 2nd Intl. Conf. AAMAS*, Melbourne, Australia, July 2003.

[21] P. E. Rybski, K. Yoon, J. Stolarz, and M. Veloso. Interactive robot task training through dialog and demonstration. In *Proceedings of the 2nd Annual Conference on Human-Robot Interaction (HRI)*, 2007.

[22] L. M. Saksida, S. M. Raymond, and D. S. Touretzky. Shaping robot behavior using principles from instrumental conditioning. *Robotics and Autonomous Systems*, 22(3/4):231, 1998.

[23] S. Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3:233242, 1999.

[24] W. D. Smart and L. P. Kaelbling. Effective reinforcement learning for mobile robots. In *In Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3404–3410, 2002.

[25] A. Stern, A. Frank, and B. Resner. Virtual petz (video session): a hybrid approach to creating autonomous, lifelike dogz and catz. In *AGENTS '98: Proceedings of the second international conference on Autonomous agents*, pages 334–335, New York, NY, USA, 1998. ACM Press.

[26] A. L. Thomaz, G. Hoffman, and C. Breazeal. Reinforcement learning with human teachers: Understanding how people want to teach robots. In *Proceedings of the 15th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN06)*, 2006.

[27] C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8(3):279–292, 1992.

[28] T. Zalla, E. Koechlin, P. Pietrini, F. Basso, P. Aquino, A. Sirigu, and J. Grafman. Differential amygdala responses to winning and losing: a functional magnetic resonance imaging study in humans. *European Journal of Neuroscience*, 12:1764–1770, 2000.