Auton Agent Multi-Agent Syst (2010) 20:85–104
DOI 10.1007/s10458-009-9095-8

Teaching a pet-robot to understand user feedback through interactive virtual training tasks

Anja Austermann · Seiji Yamada

Published online: 10 May 2009

Abstract In this paper, we present a human-robot teaching framework that uses "virtual" games as a means for adapting a robot to its user through natural interaction in a controlled environment. We present an experimental study in which participants instruct an AIBO pet robot while playing different games together on a computer generated playfield. By playing the games and receiving instruction and feedback from its user, the robot learns to understand the user's typical way of giving multimodal positive and negative feedback. The games are designed in such a way that the robot can reliably predict positive or negative feedback based on the game state and explore its user's reward behavior by making good or bad moves. We implemented a two-staged learning method combining Hidden Markov Models and a mathematical model of classical conditioning to learn how to discriminate between positive and negative feedback. The system combines multimodal speech and touch input for reliable recognition. After finishing the training, the system was able to recognize positive and negative reward with an average accuracy of 90.33%.

Keywords User feedback · Multimodality · Conditioning · Human–robot-interaction · Machine learning · Hidden Markov Models · Training tasks

1 Introduction

In recent years, a lot of research has been done focusing on creating robots that are able to communicate with humans and learn from humans in a natural way [6,7,10,14]. When

A. Austermann (✉)
The Graduate University for Advanced Studies (SOKENDAI), 2-1-2 Hitotsubashi, Chiyoda, Tokyo 101-8430, Japan
e-mail: anja@nii.ac.jp

S. Yamada
National Institute of Informatics, The Graduate University for Advanced Studies (SOKENDAI), 2-1-2 Hitotsubashi, Chiyoda, Tokyo 101-8430, Japan
e-mail: seiji@nii.ac.jp

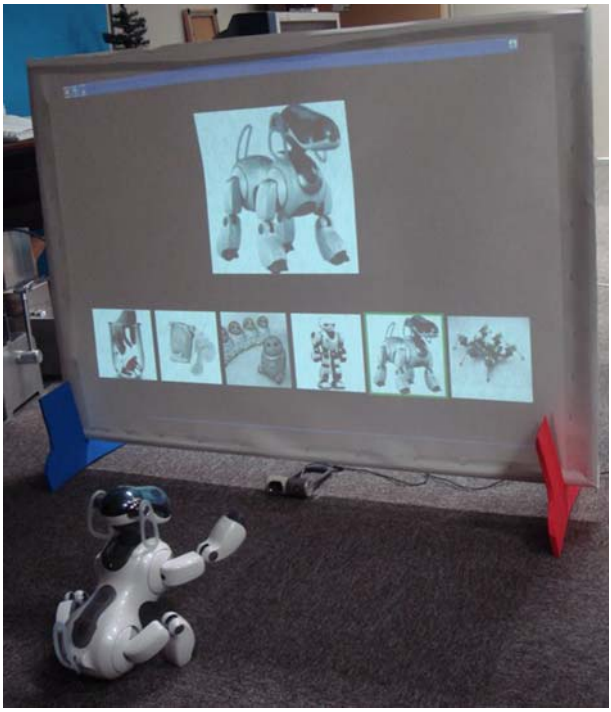

Fig. 1 AIBO During task execution

teaching a robot in a natural environment, many issues have to be handled that are not directly related to the interaction with a human, but to perceiving and modeling the environment as well as moving around and manipulating objects. Even apparently simple tasks like picking up objects cause considerable implementation effort. Using a robot simulation or a virtual agent can be an alternative in many cases but has the disadvantage that interaction cannot be perceived through the actual sensors of the robot and does not occur in the same spatial context as with a real robot. Moreover, especially in case of gesture or touch, user behavior depends on inherent properties of the robot like its size and the location of its sensors and can be expected to differ significantly between interacting with a real robot and a computer simulation.

For these reasons we implemented a client-server based framework for teaching a real robot using "virtual" tasks. A virtual task is a computer-generated visual representation of a task, in which all relevant information can be accessed and controlled directly without additional effort for implementing perception and physical manipulation of the environment. We present an experimental study that uses virtual games to allow an AIBO pet robot [6] to learn to understand multimodal positive and negative feedback from a human through natural interaction. The setting is shown in Fig. 1. The computer generates an image of a playfield, such as a "Connect Four" board or a set of "Pairs" cards, which is then projected from the back onto a white screen in front of the robot. The robot uses motion and sound to show the user which moves it is making. The use of virtual games allows us to create a controlled environment for learning a user's preferred methods of giving positive and negative feedback. By querying the game server, the robot is able to instantly assess whether a certain move is

good or not. This enables it to anticipate whether it will receive positive or negative reward for a move and to provoke and explore its user's typical feedback behavior by deliberately making good or bad moves. The game tasks are explained in detail in Sect. 4.

We chose understanding reward as a first step toward learning more general commands, because understanding whether an action has been correct or incorrect through human feedback is one of the capabilities that a robot usually needs when learning through interaction with a human instructor. In most existing service- or entertainment robot platforms, the means of giving reward to a robot are hard-coded such as predefined commands, buttons that have to be pressed or GUI-items of a remote application that have to be used for input. The user has to read a handbook and remember the correct way of giving commands and feedback. In order to enhance the user experience and to make interacting with a robot more accessible e.g., for aged people with memory deficits it is desirable to shift the effort of learning and remembering the correct way of interacting from the user to the robot.

We propose a two-staged learning method for adapting the robot to its user's feedback. In the first stage, Hidden Markov Models are used to learn to discriminate different perceptions in an unsupervised way. In the second stage, associations are learned between the trained HMMs and either positive or negative reward based on a mathematical model of classical conditioning. In the same way as previous approaches [7,9,14] to learning spoken words and symbol grounding, our algorithm attempts at assigning meanings to observations. However, our system is not trying to learn one-to-one relationships of individual words or symbols to real-world objects but focuses on relating different observations to the concepts of positive or negative feedback. These observations can be words but also touches, utterances consisting of multiple words or non-word utterances as well as combinations of speech and touch. Moreover, our proposed approach is not limited to a single modality but tries to integrate observations from different modalities.

What makes learning rewards more difficult than learning, for example, object names is that reward utterances are quite variable, often not pronounced clearly and often contain strongly emotional speech or non-word utterances like "aaah" or "ooh", which are hard to recognize by typical speech recognition systems. Details of the learning method are given in Sect. 5.

We implemented a framework for conducting experiments using the above-described virtual tasks. The architecture of the framework is described in Sect. 3. It is designed with a focus on extensibility so that it can easily be adapted to new tasks, different robots as well as virtual agents. The learning algorithm itself is independent from the concrete robot or agent used in the training.

We conducted an experimental study in order to assess how humans give feedback to a robot in a virtual game task and analyzed the observed reward behavior. Details are presented in Sect. 6. We found that the two most important modalities for giving rewards are speech and touch, while gestures were mainly used for giving instructions, not reward. We also asked the users to answer a questionnaire about their experience during the experiments to find out which features of a training task are important for successful and enjoyable teaching. With our learning method, an average recognition accuracy of 90.33% is reached for discriminating between positive and negative reward based on speech and touch.

2 Related work

Approaches to combine actual robots with virtual or mixed reality have mainly been researched upon in the field of telerobotics. However, due to the distance between the robot

and the user in a telerobotics scenario, the modalities used for interaction typically differ from the ones used in face-to-face communication. The most closely related work from the field of telerobotics was done by Xin and Sharlin [16]. They used a mixed-reality implementation of the classic Sheep and Wolves game. The sheep is a virtual, computer generated object and has to be chased by a team of four robotic wolves on a real playfield. The human is part of the robot team and interacts with the robots. The user does not have direct contact with the robots but observes the playfield through an online mixed-reality system showing the current situation on the playfield. However, interaction is not done by physically interacting with the robots but from a distance through a text-based interface. Our work focuses on modalities that are naturally used when interacting with a robot in close distance, such as speech and touch.

Another related research field is the acquisition of speech and especially the grounding of vocabulary [4,7] through human-robot interaction.

Steels and Kaplan [14] developed a system to teach the names of three different objects to an AIBO pet robot. They used so-called "language games" for teaching the connection between visual perceptions of an object and the name of the object to a robot through social learning with a human instructor. Iwahashi described an approach [7] to the active and unsupervised acquisition of new words for the multimodal interface of a robot. He applies Hidden Markov Models to learn verbal representations of objects, perceived by a stereo camera. The learning component uses pre-trained HMMs as a basis for learning while the robot interacts with its user in order to avoid and resolve misunderstandings. Kayikci et al. [9] used Hidden Markov Models and a neural associative memory for learning to understand short speech commands in a three-staged recognition procedure. First, the system recognizes a speech signal as a sequence of diphones or triphones. In the next step, the sequences are translated into words using a neural associative memory. The last step is also based on a neural associative memory. In this step, a semantic representation of the utterance is obtained.

Human reward can be used as a basis for learning in robots or virtual agents. One method for dealing with feedback from a human, that has been actively researched, is reinforcement learning with a human teacher. Kim and Scassellati implemented a method [10] to recognize approval and disapproval in a Human-Robot teaching scenario and use it to refine the robot's waving movement by Q-Learning. They use a single-modal approach to discriminate between approval and disapproval based on prosody. Lockerd et al. described an experimental setting for assessing human reward behavior and its contingency [15] for interacting with the virtual character Sophie. The participants of the study could give positive as well as negative reward to teach Sophie to bake a cake in the "Sophie's World" scenario. Reward could be given by an interactive reward interface that allowed the user to assign any reward on a scale from -1 to $+1$ either to a certain object or to the world state. They found evidence that users do not only use positive feedback to give reward to the robot after a correct move but also use it as a guidance for the robot, for instance to draw its attention to certain objects. They describe a variant of the Q-Learning algorithm, that takes into account this kind of positive feedback which is used as a guidance. Blumberg et al. [5] described how to teach a dog-like virtual character by clicker training based on reinforcement learning. Clicker training is a method which is commonly used for animal training. Their study focused on adapting reinforcement learning to work with a human teacher and on adequately modelling the state- and action-spaces for reinforcement learning.

Our learning method uses classical conditioning to establish associations between the user's utterances or touches and positive or negative reward. Classical conditioning models the learning and forgetting in animals as well as humans and was first described by Pavlov in [12]. Mathematical theories of classical conditioning were extensively researched upon

in the field of cognitive psychology. An overview can be found in [3]. One application of classical conditioning to teach an AIBO robot has been described by Yamada et al. [17]. In the study classical conditioning was used to learn which behaviors to execute in response to certain stimuli given by the user.

3 Framework design and implementation

The focus of the actual implementation of the system was to develop a framework for conducting experiments that is easy to extend and to adapt to new tasks as well as to different robots or virtual agents.

An overview of the framework is shown in Fig. 2. It is implemented using a client-server based architecture consisting of four components which communicate via TCP/IP.

- *The game server* provides the display and handling of the playfield, an evaluation function for the robot's moves as well as the opponents' artificial intelligence in case of a game for multiple players. It can use a projector or a computer screen to display the playfield.
- *The perception server* records and processes audio and video data of the user's interaction. It receives data from the robot's touch sensors, video data from two Logitech Fusion web cameras as well as audio data from a wireless lavalier microphone that is attached to the user's clothes. The data from different modalities is synchronized and stored, while the information, which is extracted from the audio and video data streams is sent to the robot control software. Learning to interpret the user's behavior using the method described in Sect. 5 takes place in the perception server.
- *The robot control software* is connected to the game server as well as the perception server and uses information about the game state to calculate the next moves of the robot. Moreover, it uses information from the perception server in order to assess whether interaction has been perceived in order to react appropriately. The AIBO Remote Framework [1] is used by the robot control software for wireless control of the robot and for reading its sensor data. Parts of this component depend on the AIBO robot and on how to send

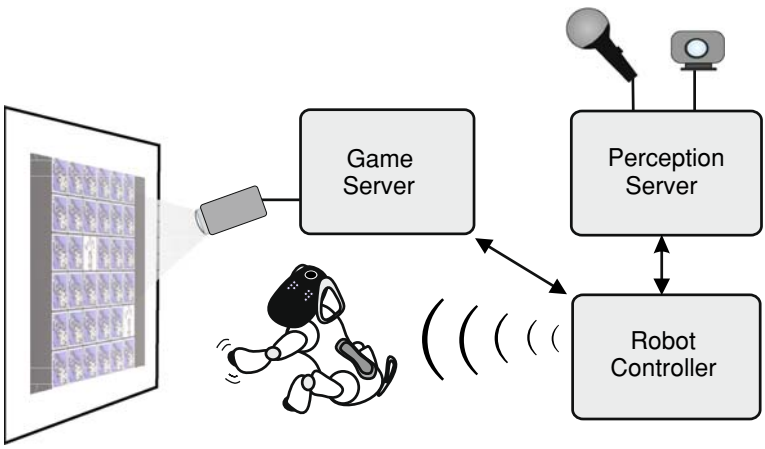

Fig. 2 Architecture of our experimental framework

commands to it and retrieve data from it. They need to be modified when using other robots or agents with our learning framework.
– *The AIBO robot itself*. We use an AIBO ERS-7 [6] for our experiments. Aibo is a dog-shaped robot which has roughly the size of a cat. It possesses twenty degrees of freedom and has touch sensors on its head and back as well as under each of its feet. Moreover, it is equipped with a camera as well as stereo microphones.

4 The training tasks

During the experiments, the image of the playfield is generated by a computer and projected from the back to the physical playfield, as seen in Fig. 1. When the robot makes a move, it points to the appropriate position on the screen and makes a sound to ask the user for feedback. It reacts to the moves of its computer opponent by looking at the appropriate positions on the playfield. In order to make the moves of the robot easier to understand for the user, the game server also visualizes them on the playfield using appropriate animations (e.g., showing a frame around the picture that AIBO is currently looking at).

Deliberately provoking positive and negative rewards from a user is only possible for the robot within a task where the human and the robot have the same understanding of which moves are desirable or undesirable. As the robot does not actually understand commands from its user at the beginning of the task, the user's commands as well as positive and negative feedback need to be reliably predictable from the task-state. If the task fulfills this condition the robot can easily explore the user's reward behavior by performing in a good or bad way.

Although the combination of Hidden Markov Models and classical conditioning is designed to be robust against occasional false training examples it is desirable to keep their number as low as possible. In order to ensure that a good move of the robot receives positive reward and a bad move receives negative reward from the user the games used for training must be designed in a way that the user can easily evaluate the situation. We assess the suitability of the different training tasks in the experiments described in Sect. 6.

4.1 Advantages of virtual training tasks

Using virtual training tasks as a basis for human-robot-communication has different benefits. As mentioned at the beginning of this paper, one main advantage is the reduction of effort needed to implement perception and understanding of the environment, so that priority can be given to the system capabilities that are actually needed for interacting with a human.

Many commercially available robots used in research such as the AIBO [6] or Khepera [11] are quite small and have no or very simple actuators. So their ability to actually manipulate objects in their environment is often quite limited. AIBO, the robot used in our experiments, can only pick-up small cylindrical objects with its mouth and needs to approach them extremely precisely in order to be able to pick them up.

Another difficulty in real-world tasks is to detect errors during task-execution such as failing to pick up an object, hitting any objects that are in the way etc. Failing to detect that an attempted action could not be performed successfully poses a risk for misinterpreting the current status of the task and misunderstanding the user's feedback.

For these reasons, we decided to implement the training task in such a way that the robot can complete it without having to directly manipulate its environment. When using a computer-based task, the current situation of the robot can be assessed instantly and correctly

by the software at any time. It can be manipulated freely, e.g., to ensure exactly the same conditions for all participants in an experiment.

4.2 Selected game tasks

The following tasks were selected to be used in our experiments, because they are easy to understand and allow a user to evaluate every move instantly. We selected four different tasks in order to see whether different properties of the task, such as the possibility to provide not only feedback but also instruction, the presence of an opponent or the game-based nature of the tasks influence the user's behavior. We implemented them in such a way that the robot is not required to perform much time-consuming walking. The four training tasks were selected so that they cover two dimensions which we assume to have an impact on the interaction between the user and the robot.

- *Easy - Difficult*: Training tasks can range from ones, that are very easy to understand and evaluate for the user, to tasks where the user has to think carefully to be able evaluate the moves of the robot correctly.
- *Constrained - Unconstrained*: In the most constrained form of interaction in our training tasks, the user is told to only give positive or negative feedback to the robot but not to give any instructions. In an unconstrained training task, the user is only informed about the goal of the task and asked to give instructions and reward to the robot freely.

The positions of the different tasks in the two dimensions can be seen in Fig. 3. There is one task for each of the combinations "easy/constrained", "easy/unconstrained" and "difficult/constrained". The reason, why there is no task for the combination "difficult/unconstrained" is that that in such a situation, the user behavior becomes too hard to predict, so that the robot cannot reliably anticipate positive or negative reward. Screenshots of the playfields can be seen in Fig. 4.

4.2.1 Find same images

On the easy/unconstrained end of the scale, there is the "Find Same Images" task. In this task, the robot has to be taught to chose the image, that corresponds to the one, shown in the center of the screen, from a row of six images. While playing, the image that the robot is currently looking or pointing at is marked with a green or red frame to make it easier for the user to understand the robot's viewing or pointing direction. By waving its tail and moving its head the robot indicates that it is waiting for feedback from its user. In this task the user can evaluate the move of the robot very easily by just looking at the sample image and the currently selected image.

The participants were asked to provide instruction as well as reward to the robot freely without any constraints to make it learn to perform the task correctly. The system was implemented in such a way that the rate of correct choices and the speed of finding the correct image increased over time.

4.2.2 Pairs

As an easy/constrained task, we chose the "Pairs" game. In this task, the robot plays the classic children's game "Pairs": At the beginning of the game, all cards are displayed upside down on the playfield. The robot chooses two cards to turn around by looking and pointing at

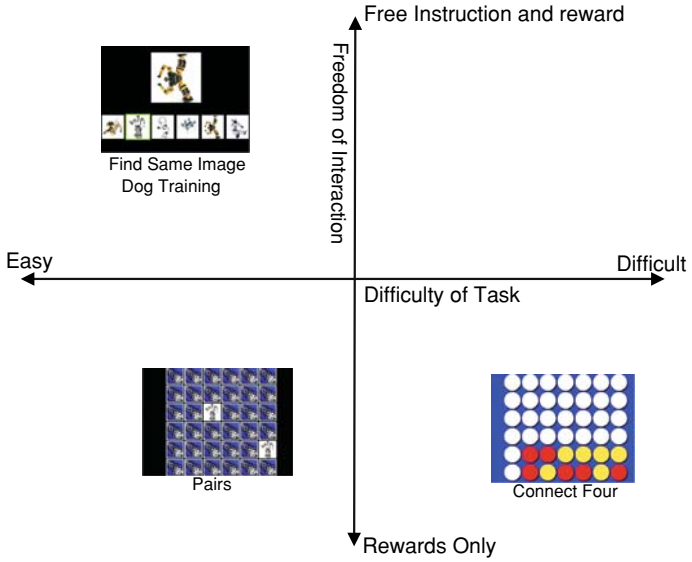

Fig. 3 Dimensions for game tasks

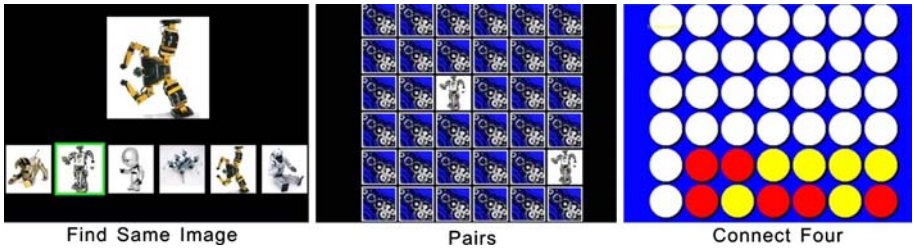

Fig. 4 Screenshots of the virtual game tasks

them. In case, they show the same image, the cards remain open on the playfield. Otherwise, they are turned upside down again. The goal of the game is to find all pairs of cards with same images in as little draws as possible. In this task the user can evaluate easily whether a move of the robot was good or bad by comparing the two selected images.

The participants were asked not to give instruction to the robot, which card to chose but to assist the robot in learning to play the game by giving positive and negative feedback only.

4.2.3 Connect four

As a difficult/constrained task, we selected the "Connect Four" task. In the "Connect Four" game, the robot plays the game "Connect Four" against a computer player. Both players take turns to insert one stone into one of the rows in the playfield, which then drops to the lowest free space in that row. The goal of the game is, to align four stones of one's own color either vertically, horizontally or diagonally.

The participants were asked to not to give instructions to the robot but provide feedback for good and bad draws in order to make the robot learn how to win against the computer player. Judging whether a move is good or bad is considerably more difficult in the "Connect

Four" task than in the three other tasks as it requires understanding the strategy of the robot and the computer player.

4.2.4 Dog training

We have implemented the "Dog Training" task as a control task in order to detect possible differences in user behavior between the virtual tasks and "normal" Human–Robot-interaction. Like the "Find Same Images" task it covers the dimensions easy/unconstrained. The user can easily evaluate the robot's behavior and use his/her way of giving instruction and reward freely without restrictions. In the "Dog Training" task, the participants were asked to teach the speech commands "forward", "back", "left", "right", "sit down" and "stand up" to the robot. The "Dog Training" task is the only task that is not game-like and does not use the "virtual playfield". Only in this task the robot was remote-controlled to ensure correct performance.

5 The learning and recognition method

We propose a learning method consisting of two stages to allow the system to adapt to the user's way of giving positive as well as negative feedback. It combines an unsupervised low-level learning stage based on Hidden Markov Models (HMMs) with a supervised learning stage based on a mathematical model of classical conditioning. In the low-level learning stage, the "reward recognition learning", the system trains HMMs to match perceived utterances. In the high-level learning stage, the "reward association learning", the system creates associations between the trained models and either positive or negative rewards. Figure 5 shows a possible result of our learning algorithm.

In this paper we are presenting results of our learning algorithm for understanding speech (utterances) and touch, combining the data from these two modalities for reliable recognition. Extensions are currently under development to deal with gesture as well as prosody of human

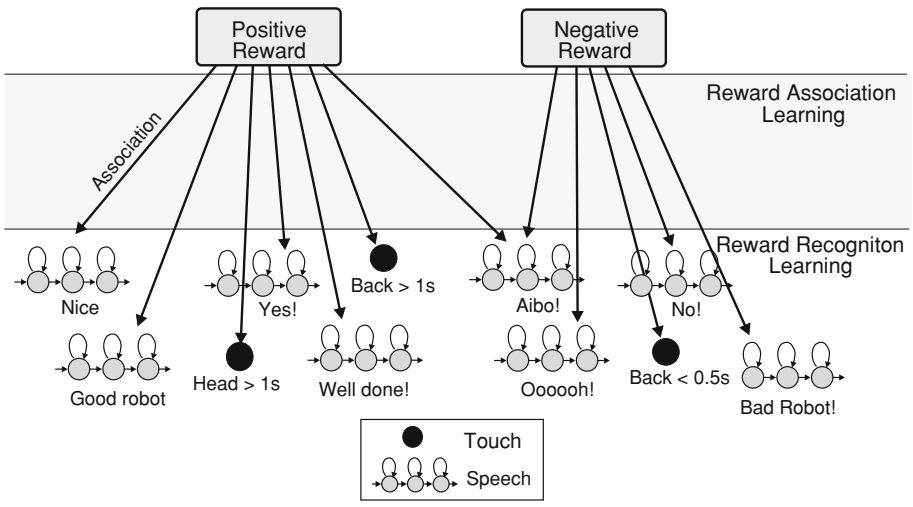

Fig. 5 Models of positive and negative reward generated by the learning algorithm

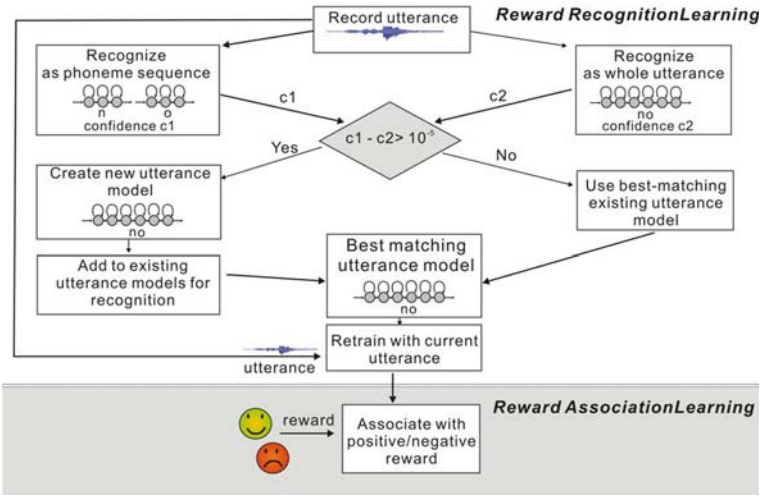

Fig. 6 Flowchart of the two stages of the learning algorithm for speech

speech. Different aspects were considered when choosing the combination of HMMs and classical conditioning for the purpose of learning to understand human feedback: By combining unsupervised clustering of similar perceptions with a supervised learning method, such as classical conditioning, our system can learn the meaning of feedback from the user during natural interaction because the learning algorithm does not require any explicit information, such as transcriptions of the user's utterances or gestures. It only needs the information whether an utterance means positive or negative feedback. This information can be determined directly from the state of the training task.

HMMs usually show high performance for the classification of time series data and are therefore widely considered state-of-the-art for this purpose. Although HMMs are typically trained in a supervised way, different approaches for an unsupervised training of HMMs have been described in literature [8].

We chose conditioning as a biologically inspired approach which typically converges quickly and has other desirable properties, which are described in Sect. 5.2. Classical conditioning allows the system to weight and combine user inputs in different modalities according to the strength of their association toward positive or negative reward for reliable recognition. An overview of the learning algorithm that is used to train the HMMs and associations is shown in Fig. 6. It is described in detail in Sects 5.1 and 5.2.

5.1 Reward recognition learning

The basis of the reward recognition learning are sets of pre-trained elementary Hidden Markov Models (HMMs) as well as a model of possible touch patterns. HMMs based on Mel-Frequency-Cepstrum-Coefficients (MFCC) are employed for the low-level modeling of speech perceptions. MFCCs are the most commonly used feature-set for HMM-based speech recognition. The extraction and use of MFCCs is described in [18].

The initial HMM-set for learning speech-based rewards contains all Japanese monophones and is taken from the Julius Speech Recognition project [19]. We use standard left-right HMMs for recognition. We decided to use monophone models instead of diphone or triphone

models although the latter are more powerful and widely used in speech recognition, because of their smaller number and lower complexity. While the monophone set for Japanese contains 43 models, 7946 HMMs are contained in the Julius triphone set for Japanese. As the initial HMMs only form a basis for constructing word models and training them in a user-dependent way, perfect accuracy is not needed in this stage. Moreover, the number of states of our word models directly depends on the number of states of the concatenated elementary models, which is significantly higher for triphone models. To keep the number of necessary training utterances low, the degrees of freedom, that is the number of states and transitions, used when training the models should not grow excessively large. We use a grammar for the phoneme recognizer that permits an arbitrary sequence of phonemes, not restricted by a language dependent dictionary, which may also contain short pauses. The grammar of our utterance model allows exactly one utterance with an optional beginning or ending silence. During the training phase, utterances from the user are detected by a voice activity detection based on energy and periodicity of the perceived audio signal.

Every time a feedback utterance from the user is observed, first the system tries to recognize the utterance with both, the phoneme sequence recognizer and the recognizer for the already trained utterance models. Matching is done by HVite, an implementation of the Viterbi Algorithm included in the Hidden Markov Model Toolkit (HTK) [18]. The results of this first step of the reward recognition learning are the best-matching phoneme sequence and the best matching utterance out of the utterance models that have been generated up to that point. In addition to that, confidence levels are output by the system for both recognition results. The confidence levels, which indicate the log likelihoods per frame, calculated by HVite, are compared to find out whether to generate a new model or retrain an existing one. If the confidence level of one of the existing word models matches the utterance well enough, that is, the confidence level of the best-fitting phoneme sequence is less than 10^{-5} better than the confidence level of the best-fitting existing utterance model then the best-fitting utterance model is retrained with the new utterance.

If the confidence level of the best-matching phoneme sequence is more than 10^{-5} better than the one of the best-fitting whole-utterance model, then a new utterance model is initialized for the utterance. The new model is created by concatenating the HMMs that make up the recognized most likely phoneme sequence. The new model is retrained with the observed utterance and added to the HMM-set of the whole-utterance recognizer. So it can be reused when a similar utterance is observed. The threshold of 10^{-5} was determined experimentally, using data that was recorded with the same audio equipment but not used for training or evaluation.

As for touch-based rewards, we decided after the experiments to abandon using complex and time-consuming HMM based modeling for the time being and decided to model touch by the following three patterns for touching the head sensor and touching the back sensor.

- Touching the robot's sensor one or multiple times for less than half a second. This typically occurs when the user is hitting the robot.
- Touching the robot's sensor for more than a second one or multiple times. This typically occurs when the user is stroking the robot.
- Touch-based interaction not falling into one of the above classes.

The output of the "Reward Association Learning" is the HMM or touch-pattern that currently models the observed reward most accurately. It serves as an input for the next stage, the reward association learning, where it is associated with either positive or negative meaning.

5.2 Reward association learning

In the reward association learning stage an association between the HMM or touch pattern obtained from the reward recognition learning and either positive or negative feedback is created or reinforced. The information of whether the HMM or touch pattern should be associated with positive or negative reward is obtained from the current state of the game. If the last move of the robot was a good one, the observation is associated with positive reward. If the last move was a bad one, the observation is associated with negative reward. The result of the reward association learning phase is a matrix, which contains the associative strengths of the different stimuli toward positive and negative rewards.

Our implementation of the reward association learning phase is based on the theory of classical conditioning. It was first described by Pavlov [12] and originates from behavioral research in animals. In classical conditioning, an association between a new, motivationally neutral stimulus, the conditioned stimulus (CS), and a motivationally meaningful stimulus, the unconditioned stimulus (US), is learned [3].

5.2.1 Relevant features of classical conditioning

Classical conditioning models several phenomena that occur when an animal or human learns through conditioning, such as blocking, extinction, sensory preconditioning and second-order conditioning, that allow our system to give priority to rewards that are used most frequently, adapt to changes in reward behavior and associate rewards which often occur together.

Blocking: Blocking occurs, when a CS_1 is paired with a US, and then conditioning is performed for the CS_1 together with a second CS_2 to the same US. In this case, the existing association between the CS_1 and the US blocks the learning of the association between the CS_2 and the US. The strength of the blocking is proportional to the strength of the existing association between the CS_1 and the US. For the learning of multimodal interaction patterns, blocking is helpful, because it allows the system to emphasize the stimuli that are most relevant. For instance, if a certain user always touches the head of the robot for showing approval, and sometimes provides different speech utterances together with touching the robot, then blocking slows down the learning of the association between approval and these speech utterances if there is already a strong association between touching the head sensor and approval.

Extinction: Extinction happens, when a CS, that has been associated with a US, is presented without the US. In that case, the association between the CS and the US is weakened. This capability is necessary to deal with changes in user behavior and with mistakes, made during the training phase, such as a misunderstanding of the situation by the human and a resulting incorrect feedback.

Sensory preconditioning and second-order conditioning: Sensory preconditioning and second-order conditioning describe the learning of an association between a CS_1 and a CS_2, so that if the CS_1 occurs together with the US, the association of the CS_2 towards the US is strengthened, too. In sensory preconditioning, learning the association between CS_1 and CS_2 is established before learning the association towards the US, in second-order conditioning, the association between the US and CS_1 is learned beforehand, and the association between CS_1 and CS_2 is learned later. Secondary preconditioning and second-order conditioning are important for our learning method, as they enable our system to learn connections between stimuli in different modalities. They also allow the system to continue learning associations between stimuli given through different modalities even when it determined whether

the robot's move was good or bad, as long as new stimuli, such as new or commands are presented together with stimuli that are already known and associated to a feedback. E.g., a new positive speech feedback is uttered with a typical, known positive/negative prosody pattern.

5.2.2 The Rescorla–Wagner-model of classical conditioning

There are several mathematical theories, trying to model classical conditioning as well as the various effects that can be observed when training real animals using the conditioning principle. The models describe how the association between an unconditioned stimulus and a conditioned stimulus is affected by the occurrence and co-occurrence of the stimuli. In this study, the Rescorla–Wagner model [13], which was developed in 1972 and has served as a foundation for most of the more sophisticated newer theories is employed. In the Rescorla-Wagner model, the change of associative strength of the conditioned stimulus A to the unconditioned stimulus US(n) in trial n, $\Delta VA(n)$, is calculated as in (1).

$$\Delta V_A(n) = \alpha_A \beta_{US(n)} (\lambda_{US(n)} - V_{all(n)}) \tag{1}$$

α_A and $\beta_{US(n)}$ are the learning rates dependent on the conditioned stimulus A and the unconditioned stimulus $US(n)$ respectively, $\lambda_{US(n)}$ is the maximum possible associative strength of the currently processed CS to the nth US. It is a positive value if the CS is present when the US occurs, so that the association between US and CS can be learned. It is zero if the US occurs without the CS. In that case, $\Delta V_A(n)$ becomes negative. Thus, the associative strength between the US and the CS decreases. $Vall(n)$ is the combined associative strength of all conditioned stimuli toward the currently processed unconditioned stimulus. The equation is updated on each occurrence of the unconditioned stimulus for all conditioned stimuli that are associated with it.

One advantage of using conditioning as an algorithm for learning the associations between positive/negative reward and the user's corresponding behaviors is its rather quick convergence, depending on the learning rate.

In this study, the learning rates for conditioned and unconditioned stimuli are fixed values for each modality but can be optimized freely. They determine how quickly the algorithm converges and how quickly the robot adapts to a change in reward behavior. We used the values $\alpha_A = 0.1$ and $\beta_{US(n)} = 0.1$ for speech as well as touch.

The maximum associative strength $\lambda_{US(n)}$ for both modalities is set to one, in case the corresponding CS is present, when the US occurs, zero otherwise. The combined associative strength of all conditioned stimuli toward the unconditioned stimulus can be calculated easily by summarizing the association values of all the CS toward the US, that have been calculated in the previous runs of the reward recognition learning.

The major drawback of the Rescorla–Wagner-Model is that it is not able to model the effects of second-order-conditioning and sensory preconditioning directly. However, this issue can be dealt with by running a second pass of the Rescorla–Wagner-algorithm to learn associations between simultaneously occurring CS. In this second pass, the CS_1 serves as the US for the conditioning of CS_2. In a third pass of the algorithm, we update the relation between the US and all CS_2, that have an association to the CS_1, using the learning rates $\alpha 2_A$ and $\beta 2_{US(n)}$, which are the product of the original learning rates α_A and $\beta_{US(n)}$ and the associative strength between the CS_1 and the corresponding CS_2.

5.3 Recognition

After the training phase has been completed the trained models can be used for recognizing rewards. All rewards that occur within a given period of time after an action of the robot are combined for multimodal recognition. In the first stage of the recognition process, the speech based rewards are recognized using the trained HMM models. The touch based rewards are matched against the above described touch patterns. The result of the first stage are individual models of the utterances and touches of the user. In the second stage, the system tries to assign a meaning to the recognition results by using the trained association matrix. The strengths of the associations between the HMMs and touch patterns toward either negative and positive reward are summed up. The reward with the highest cumulative associative strength is output as the recognition result.

6 Experiments

We experimentally evaluated our training method as well as our learning algorithm. Ten persons participated in our study. All of them were Japanese graduate students or employees at the National Institute of Informatics in Tokyo. Five of them were females, five males. The ages of the participants ranged from 23 to 47. All participants have experience in using computers. Two of them have interacted with entertainment robots before. Interaction with the robot was done in Japanese. During the experiment, we recorded roughly 5.5 h of audio and video data containing 533 rewards which consisted of 2409 individual stimuli.

6.1 Instruction and experimental setting

Figure 7 shows an overview of the experimental setting. Some videos of the interaction which were taken during the experiments are shown in Fig. 8. During the experiments the participants could sit or stand next to the AIBO robot in front of a white, semi-transparent screen. The playfield was projected onto the screen from the back.

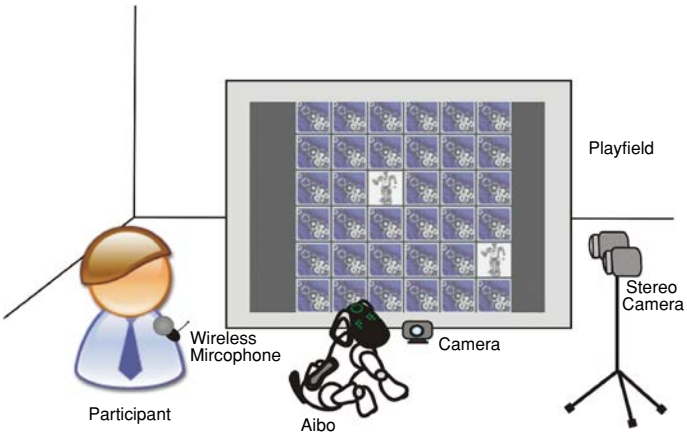

Fig. 7 Experimental setting

Fig. 8 Participants interacting with the robot in the four different game tasks

The users received some general explanations on the experiment as well as a brief explanation of every game task as a written document. The participants were asked to teach the robot, how to correctly play the different games by giving instructions and positive/negative reward for the robot's moves. They were instructed to interact with the robot naturally in their preferred way by gesture, voice and/or by touching the robots touch sensors. They received the explanation that the robot adapts to their way of teaching and learns to play the different games through their instructions.

In order to record audio and video data and to endorse the impression, that the system actually processes and learns from gestures as well as speech data a stereo camera was placed in 2.5 m distance, facing the participant, and a microphone was attached to his/her clothes. The locations of the touch sensors on the back and the head of the robot were explained to the participants. In order to give the participants the impression, that the robot is acting independently, a third dummy camera was placed below the screen facing the robot. The participants were told that it is used by the game server for recognizing the moves of the robot.

6.2 Results

We evaluated the results of our experiment with respect to the following questions:

– *Is it possible to learn multimodal user feedback in a training phase in a reasonable amount of time?* Only if reward behavior used by a single person does not vary excessively and is similar between different tasks, it can be learned effectively in a training phase and can be

recognized reliably. We have trained and evaluated our system with the data gathered in the experiments. Results concerning the recognition accuracy are presented in Sect. 6.2.1.
– *Is there a benefit in having a robot learn multimodal feedback from its user?* This is only true, if different people give reward in different ways, making it hard to handle by using hard-coded feedback patterns. Qualitative and quantitative descriptions of the feedback, given by the users, can be found in Sect. 6.2.2.
– *Which features of the training tasks are important for the user to experience an enjoyable and effective teaching situation?* We asked the participants to answer a questionnaire in order to answer this question. The results are shown in Sect. 6.2.3.

6.2.1 Recognition accuracy

We evaluated the performance of the learning algorithm offline with the data recorded within the above described experimental setting. The system was trained and evaluated with data from the "Find Same Images" and the "Pairs" task. The data from the "Connect Four" task was not used because the participants often were not able to evaluate whether a move was good or bad. Therefore reward from the user was observed for less than one third of the robot's moves in the "Connect Four" task. It had a strong positive bias and often did not match the judgment from the evaluation function of the game. We also excluded the data from the "Dog Training" task where the robot was remote-controlled. Training and evaluation were done in a user-dependent way using leave-one-out cross evaluation in order to use as much data for training and evaluation as possible. The average accuracy of our system for user-dependent classification between positive and negative rewards based on speech and touch was 90.33%. The standard deviation between users was 3.41%. As the rewards given by the participants showed a slight bias toward positive feedback, the confusion matrix, shown in Table 1 gives a more detailed overview over the performance of our recognizer. Using speech only we reached a recognition rate of 78.35% with a standard deviation of 4.37%. Using touch only the recognition rate was 76.16% with a high standard deviation of 16.92% as the usage and frequency of touch varied strongly between users. Typically one reward consists of multiple speech and touch stimuli that were given in response to one action of the robot. A stimulus is one utterance or one touch of the touch sensors. The recognition rate for individual stimuli when processing only one stimulus at a time instead of combining multiple stimuli is 80.20% with a standard deviation of 3.46%. This is about 10% lower than the recognition rate for rewards, combined from multiple stimuli, shown above. These results underline that combining stimuli given through different modalities is crucial for a reliable recognition.

6.2.2 Feedback given by the participants

As for the modalities used for giving reward, we found a strong preference for speech-based reward. Among 2409 stimuli used for giving reward, 1888 (78.37%) were given by speech, 504 (20.92%) were given by touching the robot and 17 (0.71%) were given by gestures. For the different users, the percentage of speech-based rewards ranged from 52.25% to 97.75%.

Table 1 Confusion matrix (in percent)

	Positive (actual)	Negative (actual)
Positive (recognized)	48.32	4.49
Negative (recognized)	5.18	42.01

Gestures were frequently employed by the participants for giving instructions, but we almost did not observe gestures being used for giving positive or negative reward.

Typically, multiple rewards were given for a single positive or negative behavior of the robot. Counting only the rewards given during the time, when the robot signaled that it was waiting for feedback after an action, 3.43 rewards were given for one action on average, usually including one touch reward and one to four utterances. One utterance was counted as one reward. Repetitions of an utterance were counted as multiple rewards. In case of touch reward, one or multiple contacts with the robot's touch sensors were counted as one reward, as long as the participant kept his/her hand close to the sensor.

The favorite verbal feedback differed between the users especially in case of positive reward. None of the utterances, used for positive feedback, appeared within the first six most frequently used utterances for all ten participants. On average, each person shared his/her overall most frequently used positive feedback with one other person. In case of negative reward, the feedback, given by the participants was more homogenous. The most frequently used feedback—"wrong" (chigau)—was preferred by eight out of ten persons. For the two remaining persons, it was the second and third most frequently used feedback utterance.

As for the variability of the feedback, given to the robot by an individual user: On average, participants used 12.3 different verbal expressions to convey positive feedback and 13.4 different expressions to express negative feedback. However, this number varies strongly between individuals: One person always used the same utterance for giving positive feedback and a second utterance for giving negative feedback while the person with the most variable feedback used 30 different expressions for giving positive and 28 different expressions for giving negative feedback. 55.61% of all verbal feedback was given by the participants using their preferred feedback utterance. 88.73% of a user's verbal feedback was given using one of his/her six most frequently used positive/negative utterances, so understanding a relatively small number of different utterances suffices to cover most of a participant's verbal feedback.

For positive feedback, four out of ten participants had one preferred utterance which did not vary between the four training tasks. In case of negative reward, this was true for five people. For eight out of ten participants in case of positive reward and six participants in case of negative reward, their overall most frequently used feedback utterance was among the top three feedback utterances in each individual task. In the cases, where the preferred feedback was not the same in all tasks, it typically differed for the "Connect Four" task, while in the three other tasks, including the "Dog Training" control task similar feedback was used as described above. As in the "Connect Four" task it was difficult for the users to judge, whether a move was good or bad in order to provide immediate reward, feedback tended to be very sparse and tentative like "not really good" (amari yokunai), "Is this good?" (ii kana?) or "good, isn't it" (ii deshou). A more detailed analysis of how the participants instructed the robots in the different training tasks is presented in [2].

6.2.3 Participants' evaluation of the different tasks

We prepared a questionnaire for the participants to ask about their evaluation of the different tasks. They could rate their agreement with different statements concerning the interaction on a scale from one to five, where one meant "completely agree" while five meant "completely disagree". The results can be found in Table 2. As can be seen from the table, the four tasks were considered almost equally enjoyable by the participants. For the "Find same Images" task and the "Dog Training" task, the participants' impression that the robot actually learned through their feedback and adapted to their way of teaching was strongest. Those two tasks allowed the participants to not only give feedback to the robot but also provide instructions.

Table 2 Results of the questionnaire (standard deviations given in brackets)

	Same	Pairs	Four	Dog
Teaching the robot through the given task was enjoyable	1.81 (1.04)	1.90 (0.83)	1.81 (0.89)	1.63 (0.81)
The robot understood my feedback	1.27 (0.4)	1.81 (0.74)	2.90 (0.85)	1.81 (0.30)
The robot learned through my feedback	1.36 (0.59)	2.81 (0.93)	3.45 (0.95)	1.54 (0.69)
The robot adapted to my way of teaching	1.45 (0.66)	2.63 (1.05)	3.45 (1.04)	1.64 (0.58)
I was able to teach the robot in a natural way	2.18 (0.96)	2.09 (0.86)	2.54 (1.12)	1.64 (0.69)
I always knew, which instruction or reward to give to the robot	2.00 (0.72)	2.09 (0.86)	2.90 (1.02)	1.91 (0.83)

Moreover, they were designed in a way that the robot's performance improved over time. In the "Dog Training" task, the robot was remote-controlled to react to the user's commands and feedback in a typical Wizard of OZ-Scenario. However, in the "Find Same Images" task, which was judged almost equally positively by the participants, the user's instructions and feedback were not actually understood by the robot but anticipated from the state of the training task. This did not have a negative impact on the participants impression that the robot understood their feedback, learned through it and adapted to their way of teaching. The lowest ratings were given for the "Connect Four" task. As the robot's moves could not be evaluated as easily, as in the other tasks, the participants were unsure which rewards to give and therefore did not experience an effective teaching situation. This also becomes apparent in the overall low quantity of feedback given in this task which still included incorrect feedback.

7 Conclusion

In this paper, we described and evaluated a method for learning a user's feedback for human-robot-interaction. The performance based on interpreting speech and touch rewards from a human can be considered sufficiently reliable for being used to teach a robot by reinforcement learning.

We found that natural feedback given by different users can vary strongly. Therefore, learning to understand the feedback, that a certain user employs, instead of using hard-coded and potentially unintuitive commands, which have to be learned by the user, helps to ensure natural interaction and a positive user experience.

Learning to understand feedback through a training task is only feasible and useful if the feedback given by one user is similar within different tasks. The results from the experiments suggest that this is actually the case and that typically a limited number of utterances are used by an individual to convey positive and negative reward.

However, there are cases where the contents of the utterances alone can not be correctly understood as a positive or negative reward: For instance, some of the users occasionally just repeated their previous command in a stricter tone before or instead of giving other negative feedback to the robot. In these cases, analyzing and learning the prosody, which determines the sentence melody of typical positive and negative feedback utterances, can be expected to improve the recognition accuracy. For learning to interpret commands, other than

rewards, gesture recognition will be helpful, so integrating prosody and gesture as additional modalities into our system is the current priority of our ongoing research.

Training tasks for learning to understand rewards need to be carefully designed to ensure that the robot's moves can be easily evaluated by the user. In a strategic game like "Connect Four" it is difficult to instantly assess whether a move was good or bad. This difficulty results in a decrease of the quantity as well as the correctness of the rewards and also affects the user experience.

One important question that remains open after the study is the similarity of user behavior between virtual tasks and real world tasks. Although differences in giving positive and negative reward between the virtual game tasks and the dog training task could not be observed, this does not necessarily mean that it is generally possible to train a robot for a real world task using a virtual task. This question will be targeted in a follow-up study.

References

1. AIBO remote framework. http://openr.AIBO.com
2. Austermann, A., & Yamada, S. (2008). "Good robot, bad robot" - analyzing user's feedback in a human-robot teaching task. *In proceedings of the IEEE international symposium on robot and human interactive communication 2007 (RO-MAN 08)*, (pp. 41–46).
3. Balkenius, C., & Morn, J. (1998). Computational models of classical conditioning: A comparative study. In *Proceedings of the fifth international conference on simulation of adaptive behavior*, (pp. 348–353).
4. Ballard, D.H., & Yu, C. (2003). A multimodal learning interface for word acquisition. In *Proceedings of the IEEE international conference on acoustics, speech, and signal processing*, (pp. 1520–6149).
5. Blumberg, B., Downie, M., Ivanov, Y., Berlin, M., Johnson, M.P., & Tomlinson, B. (2002). Integrated learning for interactive synthetic characters. In *Proceedings of the 29th annual conference on computer graphics and interactive techniques (SIGGRAPH 2002)*, (pp. 417–426),
6. Fujita, M., & Kitano, H. (1998). Development of an autonomous quadruped robot for robot entertainment. *Autonomous Robots, 5* (1), 7–18.
7. Iwahashi, N. (2004). Active and unsupervised learning for spoken word acquisition through a multimodal interface. In *Proceedings of the 13th IEEE international workshop on robot and human interactive communication (RO-MAN 2004)*, (pp. 437–442).
8. Li, C., & Biswas, G. (2000). A bayesian approach to temporal Data clustering using Hidden Markov models. In *Proceedings of the seventeenth international conference on machine learning (ICML 2000)*, (pp. 543–550).
9. Kayikci, Z.K., Markert, H., & Palm, G. (2007). Neural associative memories and Hidden Markov Models for speech recognition. In *Proceedings of the international joint conference on neural networks (IJCNN 2007)*, (pp. 1572–1577).
10. Kim, E.S., & Scassellati, B. (2007). Learning to refine behavior using prosodic feedback. In *Proceedings of the 6th IEEE international conference on development and learning (ICDL 2007)*, (pp. 205–210).
11. Mondada, F., Franzi, E., & Ienne, P. (1993). Mobile robot miniaturisation: A tool for investigation in control algorithms. Experimental robotics III, In *Proceedings of the 3rd international symposium on experimental robotics*, (pp. 28–30).
12. Pavlov, I.P. (1927). *Conditioned reflexes: An investigation of the physiological activity of the cerebral cortex* (G. V. Anrep, trans), New York: Oxford University Press.
13. Rescorla, R., Wagner, A. (1972). A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. In A. H. Black & W. F. Prokasy (Eds.), *Classical conditioning II: Current research and theory* (pp. 64–99). New York: Appleton Century Crofts.
14. Steels, L., & Kaplan, F. (2001). AIBO's first words : The social learning of language and meaning. *Evolution of Communication, 4*(1), 3–32.
15. Thomaz, A.L., & Breazeal, C. (2006). Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In *Proceedings of the 21st national conference on artificial intelligence (AAAI '06)*.
16. Xin, M., & Sharlin, E. (2006). Sheep and wolves: Test bed for human-robot interaction. In *CHI '06 extended abstracts on human factors in computing systems*, (pp. 1553–1558).

17. Yamada, S., & Yamaguchi, T. (2004). Training AIBO like a dog, In *Proceedings of the 13th international workshop on robot and human interactive communication (RO-MAN 2004)*, (pp. 431–436). Kurashiki, Japan.
18. Young, S. et al. (2006). The HTK book HTK version 3, http://htk.eng.cam.ac.uk/
19. The Julius speech recognition project, http://julius.sourceforge.jp/