



Learning Naturally Spoken Commands for a Robot

Anja Austermann¹, Seiji Yamada^{1,2}, Kotaro Funakoshi³, Mikio Nakano³

¹The Graduate University for Advanced Studies (SOKENDAI), Japan

²National Institute of Informatics, Japan

³Honda Research Institute Japan Co., Ltd., Japan

anja@nii.ac.jp, seiji@nii.ac.jp, funakoshi@jp.honda-ri.com, nakano@jp.honda-ri.com

Abstract

Enabling a robot to understand natural commands for Human-Robot-Interaction is a challenge that needs to be solved to enable novice users to interact with robots smoothly and intuitively. We propose a method to enable a robot to learn how its user utters commands in order to adapt to individual differences in speech usage. The learning method combines a stimulus encoding phase based on Hidden Markov models to encode speech sounds into units, modeling similar utterances, and a stimulus association phase based on classical conditioning to associate these models with their symbolic representations. Using this method, the robot is able to learn how its user utters parameterized commands, such as “Please put the book in the bookshelf” or “Can you clean the table for me?” through situated interaction with its user.

Index Terms: human-robot interaction, speech acquisition

1. Introduction

Users talk in different ways when they interact with robots. While some prefer a short, command-like style, others talk with a robot rather like with a pet or with a human communication partner. Therefore, adaptation to a user’s individual speaking style is crucial for successful, natural interaction.

We propose a learning method based on Hidden Markov Models and classical conditioning to learn how a user gives commands to a robot through situated interaction. This work is an extension of a learning method [1] [2] that we used successfully for learning to recognize user feedbacks with an average accuracy of 95.97% for recognizing positive and negative reward based on speech, prosody and touch.

However, learning commands is not as straightforward as learning feedback because commands may contain parameters and the robot needs to segment and understand the parameters correctly to understand the meaning of the command. Our system attempts to learn commands as a whole instead of parsing the utterance. The outcome of our learning method are command patterns with placeholders for parameters. For example, if the user says “Please put the ball in the box”, then “ball” and “box” are modeled as parameters while “Please put the {PAR1} in the {PAR2}” is modeled as one command pattern.

For learning commands, we use “virtual training tasks”. In these tasks, the robot interacts with the user in front of a screen and the robots’ actions are shown by gestures, speech as well as changes in the visualized scene. In the user study described in this paper, we used a living room scene to allow the robot to learn object names and commands that would be relevant for a real household robot, such as “Switch the TV on!”, or “Bring me a coffee!”. The robot is directly connected to the task server.

Based on the state of the scene the robot can precisely guess what command, object name or feedback the user is going to utter. This allows it to take over the active role in the learning process by requesting specific scenes for learning certain object names or commands from the task server. This enables the robot to systematically repeat the training of feedbacks, commands or object/place names that have not received sufficient training.

2. Related work

Various approaches towards symbol grounding and learning to understand spoken utterances have been described in literature.

Steels and Kaplan [5] 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.

Roy [3] described an approach to enable a robot to interactively learn to understand speech utterances and ground speech utterances in visual and sensory-motor experiences.

Iwahashi [4] described a method to learn to understand spoken references to visually observed objects, actions and commands which are a combination of objects and actions. Based on this knowledge the robot learned to execute the appropriate actions, that have been demonstrated by the instructor before, in response to commands from its instructor.

Gorin et al. [6] as well as Alshawi [7] proposed techniques for learning spoken language from training data without transcriptions. However, their methods were used to map whole, short utterances to non-parameterized actions. While these approaches recognize the training utterances as a whole to classify them into actions, our proposed approach is able to segment utterances into commands and their parameters. Utterances which did not occur in the training data can be recognized if they are combinations of known parameters with known commands. This allows our system to avoid combinatorial explosion when learning actions which are combinations of commands with multiple different parameters.

In contrast to previous approaches in robotic language acquisition our system tries to learn naturally spoken, domain-specific, parameterized commands that are not constrained by a restrictive grammar. The participants were instructed to utter commands to the robot in any way they consider natural, not restricting commands to simple utterances. These natural utterances typically contain non-informative words such as “please” or “can you” and there is usually no one-to-one mapping of utterances and their symbolic representation but one symbol may be represented by multiple utterances and one utterance can have multiple meanings.

The focus of this work differs from symbol grounding as

we concentrate on learning how a certain user utters commands and feedback, but assume that the robot already knows basic symbolic representations of the set of actions, that it can perform and is able to recognize objects and map them to symbols, so that it can correctly interpret a written command in simple syntax like *MOVE(BALL, BOX)*. In order to react to naturally spoken commands it needs to learn a mapping between these existing symbolic representations and commands, object names and feedback given through natural speech by the user. Having symbolic representations of objects and actions is a strong requirement, but the required mapping of objects to symbols could be performed without a human teacher or even be implemented based on markers or RFID tags attached to objects. Therefore, we decided to focus on the mapping between simple symbolic representations and users' actual utterances.

3. The training task

In principle, the proposed learning algorithm can be used with any method of gathering training data that provides spoken commands and object names as training examples along with their meanings. Transcriptions of the utterances are not necessary. In order to gather this kind of data, we implemented a training task in which the user interacts with an actual robot in front of a screen. This allowed us to create a training framework which can be used with different robots and which is extensible and can be applied to various tasks. The tasks are used to enable the robot to precisely guess the meaning of users' utterances

For learning commands, we use a "virtual living room" with various objects, such as a table, a bookshelf, a TV etc. In the first phase of the training, the robot asks the user to name these objects by pointing at an object and asking "What is that?". To make it easier for the user to understand the pointing direction of the robot, the object is highlighted on the screen. During this first learning phase the robot learns the participant's way of uttering the names of three persons and thirteen different objects in the "virtual living room".

In the command learning phase, the system learns the eight commands *MOVE(object, object)*, *SWITCH_ON(object)*, *SWITCH_OFF(object)*, *BRING(object)*, *CLEAN(object)*, *CALL(person)*, *CHARGE_BATTERY* and *SHOW_STATUS*. During command learning various changes occur in the living room scene. E.g. water is spilled on the table or it becomes dark in the room. Moreover, different possible desires of a user, such as listening to music or watching TV are visualized by thinking bubbles. The participants were instructed to respond to the changes in the scene by giving appropriate commands to the robot. Sample scenes and the symbolic representations of the expected commands are shown in Fig. 1. Learning to understand commands through virtual training tasks, instead of teaching them, for example, by demonstration enables the robot to learn commands, which would be difficult to demonstrate, such as asking the robot about its battery status or telling it to switch itself off. In a real world scenario virtual training tasks could, for example, be performed in front of the TV to familiarize the user with the functions of the robot and teach the robot to understand its user.

4. The learning method

The system learns in two successive phases. First it learns object names that can be used as parameters. Then it uses the learned object names to segment the speech signal for learning parameterized commands. In both phases, a two-

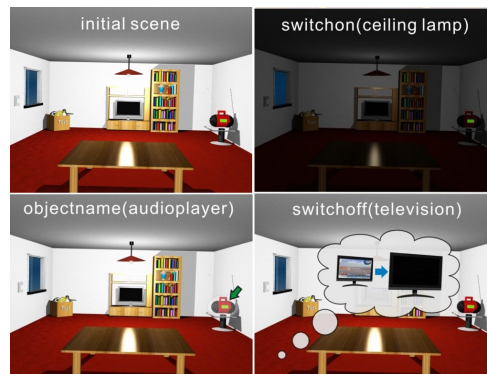


Figure 1: Sample scenes from the "virtual living room".

staged learning algorithm based on a combination of Hidden Markov Models and a mathematical model of classical conditioning is applied: The system first clusters similar sounding utterances in an unsupervised way using Hidden Markov Models in the stimulus encoding stage and then associates them with symbolic representations of object names or commands using classical conditioning in the associative learning stage. In the command learning phase, the system uses the previously learned associations between spoken object names and symbolic representations of objects. When the robot learns that the utterance "Can you switch on the TV please?" means "SWITCH_ON(TELEVISION)" it first determines, which trained HMMs for object names are associated with the parameter "TELEVISION". Then it uses these HMMs to create a grammar to find the parameters in the utterance. All parts of the utterance that are not recognized as one of the expected parameters are assumed to belong to the command itself. If there is a command model that matches the observed utterance well enough, the existing model is retrained with the utterance. Otherwise the algorithm creates a new command model from sequences of monophones modeling the parts of the utterance between the recognized parameters. An overview of the method is shown in Fig. 2.

We use the Hidden Markov Model Toolkit (HTK) [8] as well as monophone models taken from the Julius project [9] to initialize our learning algorithm. We decided to use monophone models instead of triphone models because of their smaller number and lower complexity. As the initial HMMs only form a basis for constructing utterance models which are trained with actual speech data, lower accuracy can be tolerated. Moreover, the number of states of our utterance models directly depends on the number of states of the concatenated elementary models, which is higher for triphone models. To keep the necessary amount of training utterances low, the number of states and transitions in the created utterance models should not be too large.

4.1. Stimulus encoding

The system uses two recognizers in parallel in order to determine whether an utterance is already known or whether a new utterance model should be created to encode an observed utterance. The first, phoneme-based recognizer uses monophone models when learning object names and a combination of monophone models and trained utterance models for object names when learning commands. The second, utterance-based recognizer uses only previously generated command and parameter models. It is initially empty.

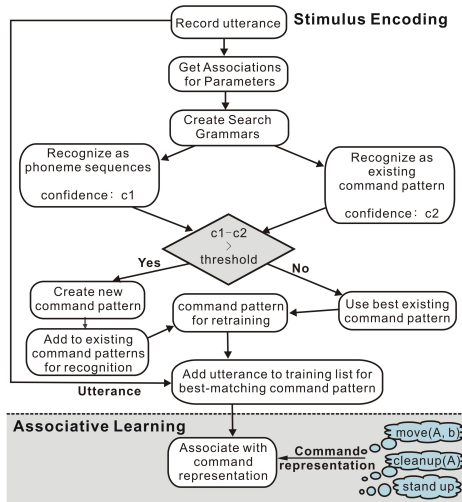


Figure 2: Overview of the learning method.

A new model is created when none of the existing models is a good match of the current utterance. This situation is detected by obtaining the recognition results of both the phoneme-based and the utterance-based recognizers and comparing the returned confidence values (log likelihood per frame). We found in previous experiments [2] that a difference in confidence of 10^{-5} is a good cutoff value to determine whether an utterance is already known or not. If the result from the phoneme sequence recognizer is not at least 10^{-5} better than the result from the utterance-based recognizer, then it is assumed that the utterance is already known and it is retrained with the current utterance. Otherwise it is assumed to be unknown and a new model is generated by concatenating the phoneme models corresponding to the recognized phoneme sequences. If a model is created for an object name or for a command which does not have any parameters, one model is created for a whole utterance. However, when learning commands with parameters, multiple models are created from all non-empty phoneme sequences before the first parameter, after the last parameter and between parameters. Information about the order of the generated utterance models as well as the parameter positions and the order of the parameters is stored separately, because this information is needed for creating the grammar for the utterance-based recognizer. For example, if the expected command is *MOVE(BALL, BOX)* and the user says “Put the ball into the box!”, then the system will create two new models, named “*move0.1*” and “*move0.2*” based on the phoneme sequences for “put the” and “into the” and stores the command pattern “*move0.1*{*PAR1*}*move0.2*{*PAR2*}”. It is possible, that a user utters a command with less parameters than expected based on the symbolic representation. For example the user might say “*It is too dark here*” to utter the command *SWITCHON(LIGHT)*. In that case the missing parameter is usually implicitly contained in the utterance. In this situation, the number of parameters in the command pattern and in the symbolic command representation do not match. This is handled in the associative learning stage of the algorithm, which is explained in section 4.2.

4.1.1. Creation of training grammars

The grammars for the recognizers, used for learning object names and command patterns, are created and updated on the

fly during the learning process. For learning object names the phoneme-based recognizer accepts an arbitrary sequence of phonemes with an optional beginning and ending silence. The utterance-based recognizer accepts exactly one known utterance model with an optional beginning or ending silence.

When learning commands, the system has to handle parameters and their positions within the utterance. Therefore search grammars for both recognizers, containing the expected parameter models, need to be created for each observed utterance. If, for example, the system observes an utterance that is assumed to correspond to the symbolic representation “*MOVE(BALL, BOX)*”, then the system first uses the trained association matrix to find all models that have an association to the symbol “*BALL*” and all models that have an association to the symbol “*BOX*”. Based on this information, a grammar for the phoneme-based recognizer is created, that allows both parameters in either order or one of the parameters or no parameter and arbitrary phoneme sequences before, between and after the parameters.

The grammar for the utterance-based recognizer contains all known command utterances and inserts the models, which are associated with the expected parameters into the positions where parameters are expected. This information is contained in the command patterns, that were generated along with the utterance models. At the position in the command pattern, where “{*PAR1*}” is found, the system inserts a nonterminal which expands to all models associated with “*BALL*”. At the position, where “{*PAR2*}” is found, the system inserts a nonterminal which expands to all models associated with “*BOX*”.

4.2. Associative learning

In order to associate the trained HMMs with their meanings, we employ the Rescorla-Wagner model [10] of classical conditioning. It is used to update an association matrix. When an utterance is recognized with a HMM sequence while an object name or command is expected, then the association between the corresponding command pattern and the symbolic representation of the command or object name is strengthened. Using classical conditioning multiple utterance models can be associated with the same symbol and vice versa. When a command is learned and the command pattern, received from the stimulus encoding, contains less parameters than the symbolic representation expects, the command pattern is not only associated with the symbolic representation of the command but with a combined symbol consisting of the command and the parameters that were expected but not found in the utterance.

4.3. Recognition

After the training, the trained models and association matrix can be used for recognizing commands. First, the utterance is recognized by the speech recognizer. It returns a sequence of recognized utterance parts. Then the system determines based on the stored command patterns which of these utterance parts are parameters and which of them belong to the command pattern. Finally, it uses the learned association matrix to determine the meaning of the command pattern or patterns. If multiple utterances occur while one command is expected, all utterances are combined to get the symbol with the highest associative strength. After the most likely meaning of the command has been determined, the system looks up the meaning of the parameters in the association matrix. The recognition result is the combination of symbolic representations of commands and objects with the highest combined associative strength to the recognized sequence of utterance parts.

The system can use available world knowledge when creating the recognition grammar. Usually not all possible parameters make sense for all possible commands. If data is available on which parameters can be used with which commands, the system can use the association matrix to create a grammar that allows only HMMs, that are associated with possible parameters to be inserted into a model sequence for a command.

5. Experiments

We evaluated the learning algorithm with data from ten participants. Five of them interacted with a pet-robot and five of them interacted with a humanoid. They used the task explained in section 3 to teach object names and parameters to the robot and were instructed to give commands and feedback naturally, using speech, touch and gestures. The experimental setting with the pet-robot is shown in Fig. 3. Every participant interacted with the robot for roughly 45 minutes until every object name and command was trained ten times. The language used in the experiments was Japanese.



Figure 3: *Experimental setting.*

We trained and evaluated the system with the recorded data using 10-fold cross evaluation and reached an average recognition accuracy of 84.45% (sd =8.23%) for recognizing the eight commands and sixteen parameters. The accuracies for the individual speakers are shown in Fig. 4. As some of the users had difficulties, interpreting some of the situations in the training tasks, the training data was checked manually, and all utterances that were clearly wrong have been removed. If only the evalua-

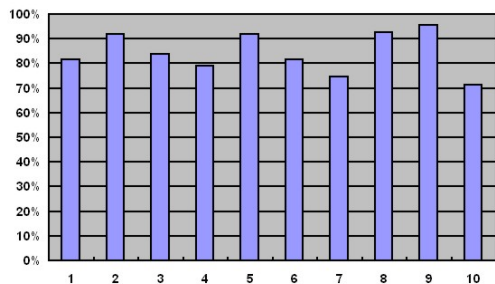


Figure 4: *Recognition rates for the participants.*

tion data was corrected and the training data was not checked or modified manually but used as recorded during the experiments, the accuracy was 80.89% (sd=7.23%). In a real world teaching scenario the raw audio data and the expected meanings are the only information available. Therefore we assumed that this way of determining the recognition rate is the best approximation of the real world performance of a robot trained with our method.

6. Discussion and conclusion

We proposed a method for learning parameterized commands for a robot. Based on a first analysis the two main reasons for misrecognitions appear to be actual user errors - e.g. misinterpretations of the situation in the training task - as well as the low amount of training data per command and the non-incremental way of training the HMMs for recognition. For persons with a very variable way of uttering commands, there were often not enough utterances available to train a HMM, so that the concatenated monophone models had to be used as they were. We expect that using incremental training or applying approaches for speaker adaptation to the generated phoneme sequences instead of simply retraining the generated HMMs with the utterances will result in an improved recognition accuracy.

Currently the system is working offline. It first records all training data and then trains the models based on the recorded data. Using incremental training of the HMMs we are planning to make the system capable of online learning during actual communication with a user.

While in this user study the system learned the participants' commands from scratch, the approach can be used easily to adapt an existing predefined set of commands to a user by only learning commands that are not matched well by the existing model. In this case the utterance-based recognizer and association matrix would not be empty upon initialization but filled with the predefined utterances.

7. References

- [1] A. Austermann, S. Yamada, "A biologically inspired approach to learning multimodal commands and feedback for Human-Robot Interaction", *CHI Work-In-Progress*, pp. 3553-3558, 2009
- [2] A. Austermann, S. Yamada, "Teaching a Pet Robot through Virtual Games", *Proceedings of the IVA '08*, pp. 308 - 321, 2008
- [3] D. Roy, "Grounded Spoken Language Acquisition: Experiments in Word Learning", *IEEE Transactions on Multimedia*, 5(2): 197-209, 2003
- [4] N. Iwahashi, "Robots that Learn Language: A Developmental Approach to Situated Human-Robot Conversation", Sanker, N. ed. *Human-Robot Interaction*, I-Tech, pp.95-118, 2007
- [5] L. Steels and F. Kaplan, "AIBO's first words : The social learning of language and meaning", *Evolution of Communication*, 4(1) pp. 3-32 , 2001
- [6] A. L. Gorin and D. Petrovksa-Delacretaz and G. Riccardi and J.H. Wright, "Learning Spoken Language without Transcriptions", *Proceedings of ASRU '99*, 1999
- [7] Hiyun Alshawi, "Effective utterance classification with unsupervised phonotactic models", *Proceedings of NAACL '03*, pp. 1-7, 2003
- [8] S. Young et al., "The HTK Book" HTK Version 3, 2006 <http://htk.eng.cam.ac.uk/>
- [9] Open-Source Large Vocabulary Continuous Speech Recognition Engine Julius: <http://julius.sourceforge.jp>
- [10] R. Rescorla, A. Wagner, "A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and non-reinforcement.", *Classical Conditioning II: Current Research and Theory*, Appleton Century Crofts, pp. 64-99, 1972