Simultaneous Action Learning and Grounding through Reinforcement and Cross-Situational Learning

Oliver Roesler
Artificial Intelligence Lab
Vrije Universiteit Brussel
Belgium
Biomedical Engineering
University of Reading
UK
oliver@roesler.co.uk

Ann Nowé
Artificial Intelligence Lab
Vrije Universiteit Brussel
Belgium
ann.nowe@vub.ac.be

ABSTRACT

Natural human-robot interaction requires robots to learn new tasks autonomously and link the learned actions to their corresponding words through grounding. Previous studies focused only on action learning or grounding, but not both. In this paper, we try to fill this gap by introducing a framework that uses reinforcement learning to learn actions and cross-situational learning to ground actions, object shapes and colors, and prepositions. The proposed framework is evaluated through a simulated interaction experiment between a human tutor and a robot. The results show that the employed framework can be used for simultaneous action learning and grounding.

KEYWORDS

reinforcement learning; cross-situational learning, symbol grounding, human-robot interaction simulation

1 INTRODUCTION

The number of service robots, which are employed in complex, human-centered environments, is growing [10, 12]. To enable them to efficiently collaborate with humans, they must be able to learn new actions autonomously and converse in natural language to understand the corresponding instructions of a user. For the former, the robot has to learn micro-action patterns, which lead to the desired changes in the environment, as specified by macro-actions[1]. For the latter, the robot has to relate words and sensory data that refer to the same characteristic of an action or object to each other, which was first described by Harnad [9] as "Symbol Grounding". Although there are many studies in the literature that investigate action learning or grounding, none of them considers both simultaneously. Additionally, action learning studies have been limited to learn a single action, such as stacking a brick onto another brick, while only varying the initial position of the gripper, due to their focus on high-dimensional action and state spaces, introduced by the use of complex grippers [8, 13]. Furthermore, grounding studies were mostly conducted offline and primarily focused on grounding of object characteristics or spatial concepts [2, 6]. Although action grounding has been considered before, the corresponding studies represented actions by simple feature vectors, which cannot be directly translated into motor commands to reproduce the original action [14, 20].

In this paper, we provide a simulation study that investigates the possibility of simultaneous action learning and grounding through the combination of reinforcement and cross-situational learning. More specifically, we simulate human-robot interactions during which a human tutor provides instructions and illustrations of the goal states of the corresponding actions. The robot then learns to reach the desired goals and grounds the words of the instructions through obtained percepts. The manipulation tasks, considered in this study, can be separated into two categories. On the one hand, tasks that move manipulation objects in regard to their initial positions and, on the other hand, tasks that move manipulation objects in regard to the position of a reference object. Additionally, objects of two diffent shapes are used, which exhibit different behaviour when manipulated, to investigate the ability of the agent to learn to execute different micro-action patterns, depending on the shape of the manipulated object. Furthermore, we investigate grounding of synonyms, i.e. words that refer to the same percepts, without using any syntactic or semantic information.
The rest of the paper is structured as follows: The next section, discusses related work on manipulation action learning as well as grounding. Section (3) provides an overview of the employed system. Section (4) describes the achieved results. Finally, Section (5) concludes the paper.

2 RELATED WORK

2.1 Action Learning

Object manipulation tasks usually require a series of actions to change the state or position of a target object [5]. Many studies have investigated how manipulation actions can be automatically learned by robots, either through demonstration or reinforcement learning [1, 8, 13, 18]. For the former, a human tutor has to demonstrate the desired action to the agent so that a policy can be derived from the recorded state-action pairs [3]. The latter, on the other hand, does not require the action to be demonstrated. Instead, it only requires a description of the goal state and discovers through trial-and-error possible policies [19]. Abdo et al. [1] proposed a method that enables robots to learn manipulation actions, such as placing one object on another, from kinesthetic demonstrations. Although, only a small number of demonstrations was necessary

[1]Micro-actions can be executed by a reinforcement learner, e.g. *move gripper left* or *close gripper* (Section 3.2), while macro-actions describe the transition from the initial state to the goal state of a situation and can be referred to by action words, i.e. verbs.

to learn the actions, the manipulator had to be directly moved by a human tutor, which might not be possible in some situations. Popov et al. [13] and Gudimella et al. [8] focused on learning to stack two objects onto each other through reinforcement learning, by directly controlling the joints of a robotic arm and gripper, which led to high-dimensional action and state spaces. The experiments were conducted in simulation due to the large number of required environment transitions.
The action, i.e. *place*, in the described studies, always resulted in the same goal position of the manipulation object with respect to the reference object. In this study, the goal position of the object can vary for the same action because of prepositions, which specify the exact goal location relative to the initial or a reference object position, thereby, illustrating the importance of investigating simultaneous action learning and grounding.

2.2 Grounding

Grounding is about the generation of meaning of an abstract symbol, e.g. a word, by linking it to perceptual information, i.e. the "real" world [9]. To ground manipulation actions in an unsupervised manner cross-situational learning can be used, which assumes that one word appears several times together with the same perceptual feature vector so that a corresponding mapping can be created [7, 16, 17]. Previous studies investigated the use of cross-situational learning for grounding of objects and actions [6, 20] as well as spatial concepts [2, 4, 21]. In all studies, grounding was conducted offline, i.e. perceptual data and words were collected in advance, which prevents their models from being used in real-time human-robot interactions. Furthermore, actions were represented through very simple or even static action feature vectors that cannot be directly used to execute the actions on a robot. Additionally, the employed models were not able to handle ambiguous words, although, the sentences humans produce are often ambiguous due to homonymy, i.e. one word refers to several objects or actions, and synonymy, i.e. one object or action can be referred to by several different words. One recent study showed that grounding of synonyms does not require semantic or syntactic information and that such information can even have a negative effect, depending on the characteristics of the used information and how it is applied [14]. Thus, for the online grounding mechanism employed in this study, no additional semantic or syntactic information is used to ground synonyms.

3 SYSTEM OVERVIEW

The employed grounding and action learning system consists of three parts: (1) Experiment simulation, which generates different situations to simulate human-robot interactions (2) Reinforcement learning algorithm, which updates Q-tables to produce optimal micro-action patterns for encountered situations, (3) Cross-situational learning component, which maps percepts to words.
The inputs and outputs of the individual parts are highlighted below, and described in detail in the following subsections.

(1) **Experiment simulation**:
   - **Output**: Situations, consisting of the initial gripper and object positions, relative goal positions of the manipulation objects, object colors, object shapes, and natural language instructions. The goal position of the manipulation object is described with respect to its initial position or the position of a reference object. The former is used in situations with one object, while the latter for situations with two objects.

(2) **Reinforcement learning**:
   - **Input**: Initial gripper position, initial object positions and the relative goal position of the manipulation object.
   - **Output**: Q-table, which produces optimal micro-action patterns for encountered situations.

(3) **Cross-situational learning**:
   - **Input**: Relative goal positions of the manipulation object, action feature vectors, object colors, object shapes, and natural language instructions.
   - **Output**: Word to percept mappings.

3.1 Experiment Simulation

During the experiment, interactions between a human tutor and a robot, in front of a tabletop, are simulated. In each situation, one or two objects, which can be of different shapes and colors, are placed on the table in different spatial configurations. If only one object is present, the instructions describe how it should be moved, e.g. *forwards* or *to the left*. If two objects are on the tabletop, the instructions determine where the manipulation object should be placed in relation to the reference object, e.g. *behind* or *on top* of it. Table (1) provides an overview of all words and phrases used in the instructions with their corresponding types and percepts. Four of the five prepositions and the action have two synonymous words, i.e. two words that refer to the same percept, thereby, allowing to investigate whether the proposed framework can handle synonyms. Prepositions are indirectly grounded through Q-tables because they directly ground relative manipulation object goal positions, which need to be represented through different Q-tables since one Q-table can only represent one goal state. Thus, the number of Q-tables is proportional to the number of prepositions, i.e. when a new preposition is encountered a new Q-table will be created. Action feature vectors represent a set of Q-tables, instead of one Q-table, because the same action word is used for different goal states, i.e. different prepositions. In this study only one macro-action is used so that also only one action feature vector exists (*AFV:AFV1*), which represents a set of five Q-tables.[2]

The experimental procedure, which is simulated in this study, consists of the following five phases:

(1) One or two objects are placed on a table and the robot determines the corresponding shapes and colors.
(2) An instruction is given to the robot by the human tutor and words and phrases are extracted.
(3) The human tutor executes the described action and the robot records the goal state, which is used to determine the desired spatial configuration, i.e. the robot does not record how the action is executed, but only the resulting state.

[2] In future work, additional macro-actions, e.g. *grab*, will be used to investigate grounding of several action feature vectors, i.e. several sets of Q-tables.

Table 1: Overview of all words and phrases used in the instructions with their corresponding types and percepts. Instructions for situations with one or two objects only differ in the used prepositions, while all other words are used in both cases.

Type	Words/Phrases		Percept
	1 Object	2 Objects	
Shape	cube		0
	ball		1
Color	red		*COLOR:red*
	green		*COLOR:green*
	blue		*COLOR:blue*
	black		*COLOR:black*
Preposition	to the left	to the left of	$[-1, 0, 0]$
	to the right	to the right of	$[1, 0, 0]$
	backwards	in front of	$[0, 1, 0$
	forwards	behind	$[0, -1, 0$
	-	on top of	$[0, 0, 1]$
Action	move		*AFV:AFV1*
	place		
Article	the		-

(4) The agent learns how to reach the goal state using reinforcement learning, thereby obtaining a corresponding micro-action pattern.
(5) Words are grounded through the obtained percepts.

In the employed simulation, the first three steps of the described experimental procedure are done simultaneously through the random generation of situations, consisting of the initial positions of the gripper and objects, the relative goal position of the manipulation object, object colors and shapes, and a natural language instruction, which describes how the manipulation object should be moved. Several constraints have been implemented to ensure that the generated situations are possible in the real world, e.g. two objects can't be at the same position. The environment is represented by a $7 \times 5 \times 2$ array so that positions are given as coordinates, i.e. $[x, y, z]$. If the gripper or an object is moved outside of the environment, a negative reward of -1 will be given and the corresponding episode will be terminated. The initial and goal positions are used to calculate the preposition percept, i.e. the relative manipulation object goal position, if only one object is present, otherwise, the goal positions of the manipulation and reference objects are used. The preposition percept only describes the direction, but not the distance, i.e. whether an object is one or two positions to the left. Object colors are words, e.g. *COLOR:red* and object shapes are numbers, e.g. *1* represents a ball.[3]
Instructions are randomly created by combining different words according to two possible structures, which are illustrated in Table (2). Examples for the first and second sentence structures are *move the red cube forwards* and *place the blue ball to the right of the black cube*. The instructions are then separated into words and phrases using a predefined dictionary.[4]

Table 2: Illustration of the two possible sentence structures, which are used depending on the number of objects, i.e. whether a reference object exists.

Position	Word/Phrase Type	
	1 Object	2 Objects
1	Action	
2	Article	
3	Manipulation Object Color	
4	Manipulation Object Shape	
5	Preposition	
6	-	Article
7	-	Reference Object Color
8	-	Reference Object Shape

3.2 Reinforcement Learning

Reinforcement learning allows an agent to learn through rewards and punishments obtained during the interaction with the environment [19]. The learning is expressed through a proper reward function, indicating the goal to the agent. In this study, the goal state is calculated via the preposition percept. This calculation needs to be done every episode because the reference object can be moved, which changes the goal position for the manpulation object. If the initial state is identical to the goal state, which can occur because the situations are generated randomly (Section 3.1), no learning takes place and the agent will continue with grounding. For each of the five possible preposition percepts (Table 1) a different Q-table is used. Q-tables are initialized with zeros and used in all situations with the given preposition percept. The number of episodes is dynamic to ensure that the agent obtains the optimal policy, independent of the difficulty of the current situation, which depends on the goal state and initial state. The dynamicity is achieved by executing Q-learning until the number of steps, required to reach the goal state, has not changed for 100 episodes because, in that case, it can be assumed that the optimal policy has been learnt[5]. Episodes are terminated when a terminal state is reached, i.e. the manipulation object is moved to its goal position or the gripper or one of the objects is moved out of the environment.
The observation vector provided to the agent contains the following information: (1) the shape of the manipulation object, (2) the gripper position relative to the manipulation object position, (3) the current manipulation object position relative to the initial manipulation object position or current reference object position, depending on whether one or two objects are present, and (4) gripper state, i.e. {open, closed}. Since the relative positions are used, the learned Q-table is applicable independent of the absolute object or gripper positions.

[3] In future work, a real robot and all five phases of the described experimental procedure will be employed. In that case, colors will be represented by RGB values and the shapes will be represented through Viewpoint Feature Histogram (VFH) [15] descriptors, which represent the object geometry taking into account the viewpoint and ignoring scale variance.

[4] In future work, words and phrases will be automatically identified in an unsupervised manner.

[5] The used critera worked for the considered situations, however, it is not optimal and might therefore be changed in the future.

The agent can execute eight different actions, which are opening or closing the gripper, moving the gripper forwards, backwards, left, or right, and lowering or raising the gripper. Physical interactions, e.g. when the gripper is moved to a position that is occupied by an object, are realistically simulated. This includes different behaviours for cubes and balls when pushed because balls will start to roll and will therefore move further than cubes. Thus, in the simulation, cubes are moved by one position and balls by two positions, unless an object occupies the second position, in which case the ball will also only be moved one position. Additionally, if the first position, to which the object is moved, is occupied by another object, both are moved.

For exploration ϵ-greedy is used as described by Sutton and Barto [19]. The exploration rate is decreased every episode, but reset for each new situation. Thus, even when the Q-table has been trained for many situations, the agent still explores many times during the first episodes of a new situation, even if a situation with the same characteristics had already been encountered before.

When the manipulation object is placed on its goal position the agent will receive a positive reward of 1. If the gripper or one of the objects is moved outside of the environment a negative reward of -1 is given. For each step a negative reward of -0.2 is given to encourage the agent to reach the goal state with the minimum number of steps possible. Additionally, potential-based reward shaping is used to reduce the number of suboptimal actions made and therefore the time required to learn [11]. The used Q-learning algorithm is represented by the following formula:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + F(s,s') + \gamma \max_{a'} Q(s',a') - Q(s,a)] \quad (1)$$

where a and a' are the actions taken in states s and s', respectively. α and γ represent the learning rate and discount factor, which are set to a value of 0.8 and 0.95, respectively. $F(s,s')$ is the potential-based reward, defined as the difference of the potential function ϕ over a source s and destination state s':

$$F(s,s') = \gamma * \phi(s') - \phi(s) \quad (2)$$

For this study the potential function ϕ is defined as follows:

$$\phi(s') = \frac{1}{\|gp(s') - mop(s')\|_1 + \|mop(s') - mop(g)\|_1 + 1} \quad (3)$$

$$\phi(s) = \frac{1}{\|gp(s) - mop(s)\|_1 + \|mop(s) - mop(g)\|_1 + 1} \quad (4)$$

where *gp* and *mop* are the positions of the gripper and manipulation object, respectively, while s and s' represent the source and destination state of the current action, and g represents the goal state.

3.3 Cross-Situational Learning

Cross-situational learning is used for grounding by creating mappings between words and percepts that occur most of the time together. Initially the set of grounded words G_w and percepts G_p is empty. After the successful execution of an action, the agent has the following perceptual information.

Algorithm 1 Grounding of words.

```
procedure GROUNDING
    Create S_{w,p}, S_{p,w}
    for j = 1 to word_number do
        Save highest P_{w,p} to G_w
    end for
    for j = 1 to percept_number do
        Save highest P_{p,w} to G_p
    end for
    return G_w ∪ G_p
end procedure
```

- Color of manipulation object.
- Shape of manipulation object.
- Relative position of manipulation object to its initial position or the position of a reference object, depending on the number of objects in the situation[6].
- Color of reference object, if a reference object is present.
- Shape of reference object, if a reference object is present.
- Action feature vector.

These perceptual information are then used together with the perceptual information of all previous situations to ground the words of all encountered instructions.[7]. Before the actual grounding procedure, all auxiliary words are discarded by checking a predefined dictionary.[8] Afterwords, a set of percepts $S_{w,p}$ is created for each word, in which each percept is saved with a number that indicates how often it occured together with that word. The same is also done for percepts, i.e. for each percept a set of words $S_{p,w}$ is created. Then, the highest word-percept pair $P_{w,p}$ is determined and saved to the set of grounded words G_w. Since w is already grounded, all pairs it is part of will not be considered for the selection of the next highest word-percept pair, during the next iteration. Additionally, the percept that was used to ground the word will not be available to ground any other words. These restrictions are applied until all percepts have been used for grounding once. If there are still ungrounded words left, all percepts will become again available for grounding, until all words have been grounded. This last step is necessary to ground synonyms. After all words have been grounded the same process is repeated for percept-word pairs $P_{p,w}$. This is necessary to assign synonymous percepts to the same word.[9] Finally, the sets of grounded words and percepts are merged. Thus, all words are mapped to all corresponding percepts. Algorithm (1) summarizes the grounding procedure.

[6]The relative position of the manipulation object is calculated by substracting the coordinates of the intial manipulation object position or reference object position from the current manipulation object position. For example, if the manipulation and reference object positions are (1, 2, 0) and (2, 2, 0), respectively, the spatial relation is $(1-2, 2-2, 0-0) = (-1, 0, 0)$.

[7]An overview of possible instructions is provided in Section (3.1).

[8]The used instructions only contain one auxiliary word, i.e. a word that has no corresponding percept, the article *the*. Since no syntactic or semantic information is used for grounding, the auxiliary word was automatically removed from the set of words given to the cross-situational learner. However, in future work, we will investigate to create the dictionary automatically in an unsupervised manner.

[9]None of the used situations contains synonymous percepts. However, they might be introduced in future work.

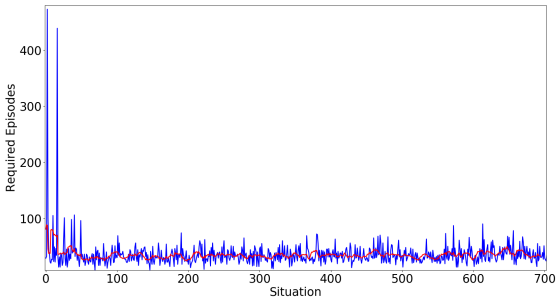

Figure 1: Number of required episodes until the reinforcement learning algorithm converges to the optimal policy for fixed initial positions (Scenario 1). The blue curve shows the original values, while the red curve shows the average.

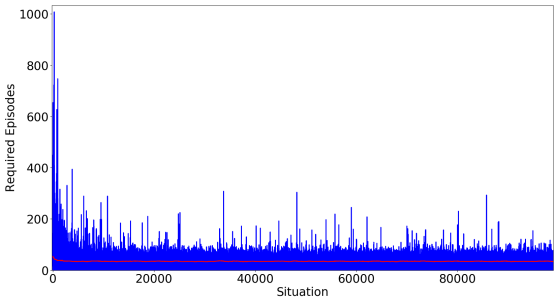

Figure 2: Number of required episodes until the reinforcement learning algorithm converges to the optimal policy for random initial positions (Scenario 2). The blue curve shows the original values, while the red curve shows the average.

4 RESULTS AND DISCUSSION

In several previous studies, reinforcement learning and cross-situational learning have been used for action learning and grounding, respectively [14, 18]. However, *to the best of our knowledge*, there has not been any study investigating simultaneous action learning and grounding. Two different scenarios are investigated. In the first scenario, each of the 704 unique instructions is used once, i.e. the number of situations is also limited to 704, and the initial gripper and object positions are always the same. The gripper is always placed in the middle of the environment, while the manipulation object is placed one position to the left of the gripper and the reference object, if present, is placed one position to the right of the gripper. For the second scenario, the initial positions are generated randomly, with the only constraint that they must be valid, e.g. two objects cannot be on the same position. In the second scenario, 100,000 different situations are generated. The following sections describe the results for the reinforcement learning as well as the cross-situational learning components for both scenarios.

4.1 Reinforcement Learning

For the first 50 situations in the first scenario, the reinforcement learner required, several times, more than 100 episodes until it converged to the optimal policy, as shown in Figure (1). Afterwards, it required on average less than 30 episodes to converge. In the second scenario, it took about 4,000 situations until it converged on average after 34 episodes to the optimal policy. At the beginning, during the first 1,000 situations, it sometimes took several hundreds up to 1,000 episodes to converge (Figures 2 and 3).

Due to the different intial positions, in the second scenario, it took longer until the number of required episodes converged to the same number as in the first scenario. That the agent did not execute the optimal policy immediately, is due to the high exploration rate at the beginning, which is always reset for each new or even already encountered situation. If the exploration rate is set to zero after a certain number of situations, the agent will execute the optimal policy in the first episode, however, if the situation has not been ecountered before, the agent will never reach the goal state.

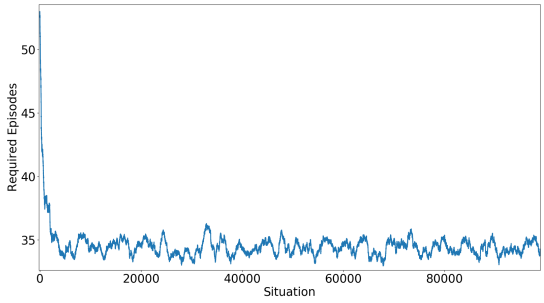

Figure 3: Average number of required episodes until the reinforcement learning algorithm converges to the optimal policy for random initial positions (Scenario 2).

4.2 Cross-Situational Learning

In the first scenario, more than 80% of the obtained groundings are false during the first situations (Figure 4). However, the performance improves quickly so that after only 3 situations 56% of the mappings are correct. After 14 situations the number of false mappings decreases to one, while the number of correct mappings still increases afterwards because of new words in subsequent situations, which have not been encountered before. Although, after 40 situations all words have been used in one or more situations, it takes 100 more situations until the last false mapping becomes correct.

For the second scenario, Figure (5) shows that more than 60% of the words are correctly grounded after 4 situations. Afterwards, the number of correct groundings increases quickly to more than 90% after 8 situations. This is followed by an increase in false groundings, due to the introduction of three new words. While two of the new words are grounded correctly after 30 more situations, it takes more than 1,900 situations until the last word is correctly grounded. Overall, the results show that the employed grounding mechanism can successfully ground all words and handle synonyms. However, only a small number of words was used in this study. Thus, it is not

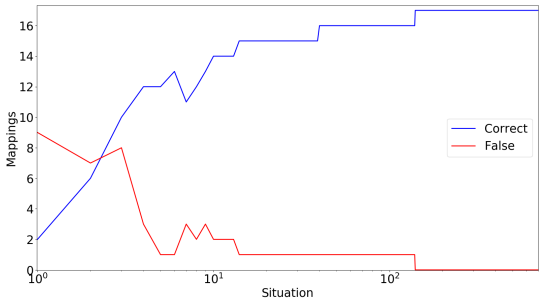

Figure 4: Cross-situational learning results for fixed initial positions. The number of correct and false mappings is shown in blue and red, respectively.

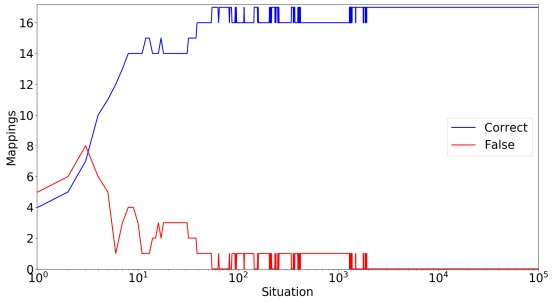

Figure 5: Cross-situational learning results for random initial positions. The number of correct and false mappings is shown in blue and red, respectively.

clear whether it would be able to handle a larger set of words since each word would be encountered fewer times. Additionally, the current grounding procedure cannot handle auxiliary words so that they need to be removed beforehand. This is not necessarily a problem, however, at the moment, the employed removal mechanism relies on a manually created dictionary.

5 CONCLUSIONS AND FUTURE WORK

We investigated a multimodal framework for simultaneous action learning and grounding of objects and actions. Our framework was set up to learn the meaning of object, action, color, and preposition words using object shapes and colors, learned micro-action patterns, and relative object positions.

The proposed framework allowed the learning of actions as well as the grounding of words, including synonyms, during a simulated human-robot interaction. However, it relies on a manually defined dictionary to identify auxiliary words and phrases. Additionally, only a small number of words has been used and the used percepts have all been represented through simple words and numbers, which is different from real sensor data.

In future work, we will use a stereo camera to obtain the shapes, colors, and positions of objects and a robot to execute learned actions. However, action learning will still be done in simulation, to speed up learning and avoid situations in which human intervention is necessary. Furthermore, we will consider automatic identification of phrases and auxiliary words. Finally, we will investigate whether the grounding mechanism works for a larger number of words.

REFERENCES

[1] N. Abdo, L. Spinello, W. Burgard, and C. Stachniss. 2014. Inferring What to Imitate in Manipulation Actions by Using a Recommender System. In *IEEE International Conference on Robotics and Automation (ICRA)*. Hong Kong, China.

[2] A. Aly, A. Taniguchi, and T. Taniguchi. 2017. A Generative Framework for Multimodal Learning of Spatial Concepts and Object Categories: An Unsupervised Part-of-Speech Tagging and 3D Visual Perception Based Approach. In *IEEE International Conference on Development and Learning and the International Conference on Epigenetic Robotics (ICDL-EpiRob)*. Lisbon, Portugal.

[3] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. 2009. A survey of robot learning from demonstration. *Robotics and Autonomous Systems* 57 (2009), 469–483.

[4] C. R. Dawson, J. Wright, A. Rebguns, M. V. Escárcega, D. Fried, and P. R. Cohen. 2013. A generative probabilistic framework for learning spatial language. In *IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL)*. Osaka, Japan.

[5] R. Flanagan, M. C. Bowman, and R. S. Johansson. 2006. Control strategies in object manipulation tasks. *Current Opinion in Neurobiology* 16 (2006), 650–659.

[6] J. F. Fontanari, V. Tikhanoff, A. Cangelosi, R. Ilin, and L. I. Perlovsky. 2009. Cross-situational learning of object-word mapping using Neural Modeling Fields. *Neural Networks* 22, 5-6 (July-August 2009), 579–585.

[7] J. F. Fontanari, V. Tikhanoff, A. Cangelosi, and L. I. Perlovsky. 2009. A cross-situational algorithm for learning a lexicon using Neural modeling fields. In *International Joint Conference on Neural Networks (IJCNN)*. Atlanta, GA, USA.

[8] A. Gudimella, R. Story, M. Shaker, R. Kong, M. Brown, V. Shnayder, and M. Campos. 2017. Deep Reinforcement Learning for Dexterous Manipulation with Concept Networks. *CoRR* (2017). http://arxiv.org/abs/1709.06977

[9] S. Harnad. 1990. The Symbol Grounding Problem. *Physica D* 42 (1990), 335–346.

[10] C. C. Kemp, A. Edsinger, and E. Torres-Jara. 2007. Challenges for Robot Manipulation in Human Environments. *IEEE Robotics & Automation Magazine* 14, 1 (March 2007), 20–29.

[11] A. Y. Ng, D. Harada, and S. Russell. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML)*, I. Bratko and S. Dzeroski (Eds.), Vol. 99. 278–287.

[12] International Federation of Robotics. 2017. World Robotics 2017 - Service Robots. (October 2017).

[13] I. Popov, N. Heess, T. Lillicrap, R. Hafner, G. Barth-Maron, M. Vecerik, T. Lampe, Y. Tassa, T. Erez, and M. Riedmiller. 2017. Data-efficient Deep Reinforcement Learning for Dexterous Manipulation. *CoRR* (2017). http://arxiv.org/abs/1704.03073

[14] O. Roesler, A. Aly, T. Taniguchi, and Y. Hayashi. 2018. A Probabilistic Framework for Comparing Syntactic and Semantic Grounding of Synonyms through Cross-Situational Learning. In *ICRA-18 Workshop on Representing a Complex World: Perception, Inference, and Learning for Joint Semantic, Geometric, and Physical Understanding*. Brisbane, Australia.

[15] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu. 2010. Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Taipei, Taiwan, 2155–2162.

[16] J. M. Siskind. 1996. A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition* 61 (1996), 39–91.

[17] K. Smith, A. D. M. Smith, and R. A. Blythe. 2011. Cross-Situational Learning: An Experimental Study of Word-Learning Mechanisms. *Cognitive Science* 35, 3 (2011), 480–498.

[18] F. Stulp, E. A. Theodorou, and S. Schaal. 2012. Reinforcement Learning With Sequences of Motion Primitives for Robust Manipulation. *IEEE Transactions on Robotics (T-RO)* 28, 6 (December 2012), 1360–1370.

[19] R. S. Sutton and A. G. Barto. 1998. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

[20] A. Taniguchi, T. Taniguchi, and A. Cangelosi. 2017. Cross-Situational Learning with Bayesian Generative Models for Multimodal Category and Word Learning in Robots. *Frontiers in Neurorobotics* 11 (2017).

[21] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. Teller, and N. Roy. 2011. Approaching the symbol grounding problem with probabilistic graphical models. *AI Magazine* 32, 4 (2011), 64–76.