The Knowledge Engineering Review, Vol. 34, e13, 1 of 14.
doi:10.1017/S0269888919000079

Action learning and grounding in simulated human–robot interactions

OLIVER ROESLER and ANN NOWÉ

Artificial Intelligence Lab, Vrije Universiteit Brussel, Pleinlaan 9, 1050 Brussels, Belgium;
e-mails: oliver@roesler.co.uk, ann.nowe@vub.ac.be

Abstract

In order to enable robots to interact with humans in a natural way, they need to be able to autonomously learn new tasks. The most natural way for humans to tell another agent, which can be a human or robot, to perform a task is via natural language. Thus, natural human–robot interactions also require robots to understand natural language, i.e. extract the meaning of words and phrases. To do this, words and phrases need to be linked to their corresponding percepts through grounding. Afterward, agents can learn the optimal micro-action patterns to reach the goal states of the desired tasks. Most previous studies investigated only learning of actions or grounding of words, but not both. Additionally, they often used only a small set of tasks as well as very short and unnaturally simplified utterances. In this paper, we introduce a framework that uses reinforcement learning to learn actions for several tasks 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 both action learning and grounding.

1 Introduction

The number of service robots that are employed in complex and human-centered environments instead of factories is growing (IFR, 2017), thereby bringing us closer to a future in which robots are an essential part of everyday life. The challenge of human-centered environments is that they cannot be controlled, especially due to the inevitable interactions of robots with untrained users, who might neither understand nor act according to the limitations of the robots. In contrast, factory environments are highly controlled and can be adjusted to the capabilities of employed robots (Kemp *et al.,* 2007). Due to the complexity, unpredictability, and dynamicity of non-industrial environments, employed service robots must be able to learn new tasks autonomously, i.e. when only the goal of the task is given without any further information. This does not mean that robots should not use further guidance by a user, such as demonstrations, but that they should still be able to learn new tasks, if no guidance is available. The goals of the tasks can be described in many different ways, e.g. through written natural language descriptions or simple pointing by a human tutor, to allow robots to learn by themselves the optimal ways of execution. Furthermore, it cannot be expected that users will learn a limited set of instructions that can be hard-coded into the robots to let them execute specific tasks. Instead, robots must be able to understand natural language instructions to accurately identify the requested tasks and determine whether they know how to execute them. Understanding instructions is non-trivial and requires connections between symbols, i.e. words used in instructions, and their meanings. The latter can, in theory, be provided by relating unknown symbols to other symbols. However, this only works, if the meaning of the other symbols is known, i.e. relating unknown symbols to other unknown symbols does not constitute meaning to the former. Thus, to provide meaning agents need mappings from words to corresponding percepts, such as color values, geometric object characteristics, or micro-action patterns. Therefore, tasks or macro-actions, which can

be referred to by a single word, i.e. a verb, such as *push*, will be described by micro-action patterns, which can be directly converted to actuator commands and thus executed by robots. In contrast, macro-actions refer to specific changes in the environment that describe the transition from an initial state to the goal state of a situation. Creating relations between words and sensory data that refer to the same characteristic of an action or object is called 'Symbol Grounding', which was first introduced by Harnad (1990). The main idea is that abstract knowledge and language only becomes meaningful, when it is linked to the physical world. However, not all words need to be directly grounded through percepts because they can be indirectly grounded by being linked to directly grounded words.

Although there are many studies in the literature that investigate action learning or grounding, only a few consider both simultaneously and they significantly differ in their approaches and experimental setups. 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 (Gudimella *et al.,* 2017; Popov *et al.,* 2017). Furthermore, grounding studies were mostly conducted offline and primarily focused on grounding of object characteristics or spatial concepts (Fontanari *et al.,* 2009a; Aly *et al.,* 2017). Although action grounding has been considered before, corresponding studies represented actions by simple feature vectors, which cannot be directly translated into motor commands to reproduce the original action (Taniguchi *et al.,* 2017; Roesler *et al.,* 2018).

In this paper, we investigate the possibility of simultaneous action learning and grounding through the combination of reinforcement learning (RL) and cross-situational learning (CSL). 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 positions of reference objects. Additionally, the agent has to take the object shapes into account because different shapes lead to different behaviors during manipulation. Therefore, different micro-action patterns need to be executed by the agent depending on the shape of the manipulated object. Furthermore, we investigate the use of CSL for unsupervised identification of auxiliary words, i.e. words that do not have a corresponding percept, and phrases, i.e. groups of words that are grounded through one percept. Finally, we examine whether the employed grounding mechanism can handle synonyms, i.e. words that refer to the same percepts, without the help of any syntactic or semantic information.

The rest of the paper is structured as follows: the next section provides some background regarding RL and CSL. Afterward, related work on manipulation action learning as well as grounding is discussed in Section 3. Section 4 provides an overview of the employed system. Section 5 describes the achieved results. Finally, Section 6 concludes the paper.

2 Background

This section provides a brief overview of RL and CSL.

2.1 Reinforcement learning

RL is a framework that allows an agent to learn how to act in a correct and optimal manner in a complex environment through the maximization of a reward signal (Sutton & Barto, 1998). The environment is defined as everything that is outside of the agent. Interactions between the agent and environment happen in a loop. First, the agent observes the current state of the environment and uses prior experience, i.e. observations of the effect of previously executed actions, to select an action. Afterward, it executes the selected action. Finally, it observes the effect of the action by observing the new state of the environment and receives a reward signal. This signal specifies the long-term effect of an action and is given either by the environment or generated by the agent itself. The latter is the case, if the agent knows its goal

https://doi.org/10.1017/S0269888919000079 Published online by Cambridge University Press

state and is able to calculate how much the distance to the goal changed through the executed action. The overall goal of an RL agent is to obtain a policy, i.e. a function that specifies which action should be taken for all possible situations. Thereby, allowing it to maximize the cumulative reward received over time.

Typically, an RL problem is modeled as a Markov decision process (MDP), which can be represented as a 4-tuple $< \mathbf{S}, \mathbf{A}, \mathbf{T}, \mathbf{R} >$, where **S** is the state space, i.e. the set of all possible states, **A** is the action space, i.e. the set of all possible actions, **T** is the transition probability function that describes the probability that action **a** in state **s** results in state $\mathbf{s}'$, and **R** is a function specifying the reward received when transitioning from state **s** to $\mathbf{s}'$ through the execution of action **a**. In an MDP, $\mathbf{s}'$ depends only on **a** and **s**, i.e. all previous actions and states have no effect (Puterman, 1994).

2.2 Cross-situational learning

CSL is a mechanism for word learning that is able to handle referential uncertainty by learning the meaning of words across multiple exposures. The basic idea, which has been proposed among others by Pinker (1989) and Fisher *et al.* (1994), is that the context a word is used in leads to a number of candidate meanings, i.e. mappings from words to percepts, and that the correct meaning lies at the intersection of the sets of candidate meanings. Thus, the correct mapping between a word and its corresponding percepts can only be found through repeated co-occurrences so that the learner can select the meaning which reliably reoccurs across situations (Blythe *et al.,* 2010; Smith & Smith, 2012). The original idea of CSL was developed to explain how humans learn words, when no prior knowledge of language is available. A number of experimental studies have confirmed that humans use CSL for word learning (Akhtar & Montague, 1999; Gillette *et al.,* 1999; Smith & Yu, 2008). CSL belongs to the group of slow-mapping mechanisms, that is, word learning mechanisms that require more than one exposure, through which most words are acquired (Carey, 1978). In contrast, *fast-mapping*, which can neither be explained nor achieved through CSL, allows words to be acquired through a single exposure, but is only employed for a limited number of words (Carey & Bartlett, 1978; Vogt, 2012). Many different algorithms have been proposed to simulate CSL in humans and enable artificial agents, such as robots, to learn the meaning of words by grounding them through percepts (Section 3.2).

3 Related work

3.1 Action learning

Object manipulation tasks usually require a series of actions to change the state or position of a target object (Flanagan *et al.,* 2006). Many studies have investigated how manipulation actions can be automatically learned by robots. Manipulation actions are high-level macro-actions that consist of a sequence of low-level micro-actions. The latter can be defined in many different ways, thereby determining which learning approaches are most appropriate. For example, micro-actions can be represented through the movements of individual joints (Gu *et al.,* 2017; Popov *et al.,* 2017), simple fine-grained movements of end effectors, or sophisticated and complex movements of end effectors or body parts, which allows the use of very high-level learning mechanisms, such as precise guidance through natural language instructions (She *et al.,* 2014). When micro-actions are represented through simple movements of joints or end effectors, most studies employed learning through demonstration or RL (Stulp *et al.,* 2012; Abdo *et al.,* 2014; Popov *et al.,* 2017; Gudimella *et al.,* 2017). 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 (Argall *et al.* 2009). 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 (Sutton & Barto, 1998). Abdo *et al.* (2014) 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 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.* (2017) and Gudimella *et al.* (2017)

https://doi.org/10.1017/S0269888919000079 Published online by Cambridge University Press

focused on learning to stack two objects onto each other through RL, 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.

3.2 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 (Harnad, 1990). There are many different mechanisms for grounding. She *et al.* (2014) investigated the use of a dialog system for grounding of higher level symbols through already grounded lower level symbols. While it can be used as an additional grounding mechanism, its usefulness is limited due to the need for a sufficiently large set of grounded lower level symbols. Additionally, the system requires a professional tutor to answer its questions, who might not always be available and increases the cost to obtain new groundings. The latter problem also constraints the applicability of the *Naming Game*, which allows an agent to quickly learn word-percept mappings, if another agent is present and knows the correct mappings (Steels & Loetzsch, 2012). To ground manipulation actions in an unsupervised manner, i.e. without the need for a tutor, CSL (Section 2.2) 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 (Siskind, 1996; Fontanari *et al.,* 2009b; Smith *et al.,* 2011). Previous studies investigated the use of CSL for grounding of objects and actions (Fontanari *et al.,* 2009a; Taniguchi *et al.,* 2017) as well as spatial concepts (Tellex *et al.,* 2011; Dawson *et al.,* 2013; Aly *et al.,* 2017). In all studies, grounding was conducted offline, i.e. perceptual data and words were collected in advance. Fontanari *et al.* (2009a) did not even present situations separately, but assumed that data of all situations are already available because it was required by their employed Neural Modeling Fields Algorithm. This 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. For example, Taniguchi *et al.* (2017) represented actions through proprioceptive and tactile features, which are obtained after the robot completes an action. 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 known synonyms, i.e. synonyms that have been encountered in previous situations, 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 (Roesler *et al.,* 2018). In contrast, another study showed that unknown synonyms, i.e. synonymous words of previously encountered words that have not been encountered before, require semantic and syntactic information to be grounded (Roesler *et al.,* 2019). Since all words appear in several situations, the online grounding mechanism employed in this study uses no additional semantic or syntactic information to ground synonyms.

4 System overview

The employed grounding and action learning system consists of three parts: (1) human-robot interaction simulation, which generates different situations, (2) RL algorithm, which learns optimal micro-action patterns for encountered situations, (3) CSL component, which identifies auxiliary words and phrases, and maps percepts to non-auxiliary words and phrases.

The inputs and outputs of the individual parts are highlighted below, and described in detail in the following subsections:

https://doi.org/10.1017/S0269888919000079 Published online by Cambridge University Press

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. Percept names are just placeholders, that is, the CSL algorithm only uses the names to distinguish different percepts.

Type	**Words/phrases**		**Percept**
	One object	**Two objects**	
Shape	cube, block, hexahedron, quadrate		0
	ball, sphere, spheroid, pellet, globe, orb, globule		1
Color	red, reddish		red
	green, greenish		green
	blue, blueish		blue
	black, blackish		black
Preposition	to the left	to the left of	[–1, 0, 0]
	to the right	to the right of	[1, 0, 0]
	backwards, toward the rear, rearward	in front of	[0, 1, 0]
	forwards, toward the front, ahead	behind	[0, –1, 0]
	up	on top of, above, over	[0, 0, 1]
Action	move, place, displace, put		AFV1
Article	the		—

1. HRI 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, where words can also be phrases.

4.1 HRI 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, for example, *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, for example, *behind* or *on top* of it. Table 1 provides an overview of all words and phrases used in the instructions with

https://doi.org/10.1017/S0269888919000079 Published online by Cambridge University Press

their corresponding types and percepts. All words have at least one synonym, i.e. a word that refers to the same percept, thereby allowing to investigate whether the proposed framework can handle synonyms. Action feature vectors are represented by Q-tables. In this study, only one macro-action is used so that also only one action feature vector (*AFV1*) and Q-table exist[1].

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 a human tutor and words and phrases are extracted.
3. The human tutor points to an object and then to its goal position. If only one object is present, the robot determines the desired spatial configuration using the initial and goal positions of the object. In contrast, if a second object is present, the spatial configuration will be determined using the goal positions of both objects.
4. The agent learns how to reach the goal state using RL, thereby obtaining a corresponding micro-action pattern.
5. Words and phrases are grounded through obtained percepts.

In the employed simulation, the first three steps of the described experimental procedure are done simultaneously through the generation of situations, which consist of the initial gripper and object positions, 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 cannot 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. *red* and object shapes are numbers, e.g. '1' represents a ball[2].

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*. Afterward, unsupervised CSL algorithms are used to identify and remove auxiliary words as well as separate instructions into words and phrases (Sections 4.3.1 and 4.3.2).

4.2 Reinforcement learning

RL allows an agent to learn through rewards and punishments obtained during the interaction with the environment (Sutton & Barto, 1998). 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 manipulation object. If the initial state is identical to the goal state, which can occur because the situations are generated randomly (Section 4.1), no learning takes place and the agent will continue with grounding. The Q-table is initialized with zeros. The number of episodes is dynamic to ensure that the agent obtains the optimal policy, independent of the difficulty of the current situation,

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

[2] 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 (Rusu *et al.*, 2010) descriptors, which represent the object geometry taking into account the viewpoint and ignoring scale variance.

https://doi.org/10.1017/S0269888919000079 Published online by Cambridge University Press

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	
	One object	**Two 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

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[3]. 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 preposition describing the goal position of the manipulation object, (2) the shape of the manipulation object, (3) the gripper position relative to the manipulation object position, (4) 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 (5) 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.

The agent can execute eight different actions, which are opening or closing the gripper, moving the gripper forward, backward, 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 behaviors 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 (1998). The exploration rate is decreased every episode, while two different strategies of how to treat exploration across situations have been investigated. On the one hand, the exploration rate is reset for each situation to ensure that the agent will be able to learn novel situations that are encountered even after several thousand situations, which can happen because situations are generated randomly. Thus, even if situations with the same characteristics have been encountered before, the agent will execute many exploratory actions during the first episodes of a new situation. On the other hand, the exploration rate is decreased continuously and shared across all situations.

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

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

https://doi.org/10.1017/S0269888919000079 Published online by Cambridge University Press

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 (Ng *et al.,* 1999). The used Q-learning algorithm is represented by the following formula:

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

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':

$$F(s, s') = \gamma * \phi(s') - \phi(s) \tag{2}$$

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

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

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

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.

4.3 Cross-situational learning

The idea of CSL has led to the development of a variety of algorithms that realize CSL in different ways, e.g. through the use of probabilistic models (Aly *et al.,* 2017; Roesler *et al.,* 2019), for grounding of words through percepts in artificial agents. In this section, three CSL algorithms are proposed, which employ CSL in a way that, to the best of our knowledge, has not been proposed or used before. The proposed CSL algorithms are not only used to identify words and percepts that occur most of the time together so that corresponding mappings can be created that ground words through percepts, but also to detect auxiliary words and phrases.

Initially, the set of grounded words (GW) and percepts (GP) is empty. After the successful execution of an action, the agent has the following perceptual information.

- 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[4].
- 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[5]. Before the actual grounding procedure, auxiliary word and phrase detection procedures are applied to identify and discard auxiliary words, and identify phrases so that they can be treated as one word for grounding, i.e. all words of a phrase are

[4] The relative position of the manipulation object is calculated by subtracting the coordinates of the initial 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).

[5] An overview of possible instructions is provided in Section 4.1.

https://doi.org/10.1017/S0269888919000079 Published online by Cambridge University Press

Algorithm 1 The procedure to update sets of permanent mappings (PM) and auxiliary words (AW) takes as input all words (W) and percepts (P) of the current situation and returns updated PM and AW.

procedure UPDATE PERMANENT MAPPINGS AND AUXILIARY WORDS(W, P)
    Add words that occur at least twice in W to set $W2$
    Add percepts that occur at least twice in P to set $P2$
    if $|W2| > 0$ and $|P2| == 0$ **then**
        Add $W2$ to AW
    else if $|W2| == 1$ and $|P2| == 1$ **then**
        Add $W2$ and $P2$ to PM
    end if
    return AW, PM
end procedure

grounded by the same percept. The proposed grounding procedure can be separated into three independent parts, i.e. three different CSL algorithms, as outlined below and described in detail in the following subsections.

1. Detection of permanent mappings and auxiliary words (Section 4.3.1).
2. Detection of phrases (Section 4.3.2).
3. Grounding of non-auxiliary words and phrases (Section 4.3.3).

4.3.1 Auxiliary word detection

Auxiliary words are detected in an unsupervised manner using CSL, i.e. the detection performance improves with the number of encountered situations. If a word occurs more than one time in a given situation and all percepts only occur once, the word will be added to the set of auxiliary words (AW). In contrast, when one word and one percept occur several times, they will instead be added to the set of permanent mappings (PM) because it is a clear indication that the word is grounded by the percept. The mechanism is illustrated in Algorithm 1.

4.3.2 Phrase detection

Phrases are identified with the help of weighted n-grams. First, all n-grams of the current instruction are identified and added to the set of all n-grams (NG) obtained from all encountered instructions. Afterward, all n-grams containing words that are in AW or PM are removed. Then, the weight of each n-gram is calculated by adding the number of occurrences of all words of the n-gram together and dividing the result by the number of words. Afterward, the confidence score is calculated for each n-gram by dividing its weight by the sum of the weights of all n-grams. Finally, all n-grams that have a score greater than a predefined threshold are added to the set of permanent phrases (PP). Once a phrase has been added to PP, it cannot be removed. Algorithm 2 summarizes the phrase detection procedure.

4.3.3 Grounding

To ground all words and phrases, a set of percepts is created for each word (WP), in which each percept is saved with a number indicating how often it occurred together with that word. The same is also done for percepts, that is, for each percept a set of words is created (PW). Then, the highest WP is determined and saved to the set of grounded words (GW). All other WP the word is part of will not be considered for the selection of the next highest WP during the next iteration because it is already grounded. 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

https://doi.org/10.1017/S0269888919000079 Published online by Cambridge University Press

Algorithm 2 The phrase detection procedure takes as input all words (W) of the current situation and the sets of previously obtained n-grams (NG), auxiliary words (AW), permanent mappings (PM) and permanent phrases (PP) and returns an updated PP.

procedure PHRASE DETECTION(List of words)
  Identify n-grams and add to NG
  Remove n-grams from NG that contain words that are in AW or PM
  Weight n-grams: $weight = \frac{\sum_{i=0}^{n} wordOccurrenceCount[i]}{n}$
  Obtain score for each n-gram: $score = \frac{weight}{\sum_{i=0}^{n} weight[i]}$, where n is the number of n-grams.
  if $score > 0.1$ then
    Add n-gram to PP
  end if
  return PP
end procedure

Algorithm 3 The grounding procedure takes as input all words (W) and percepts (P) of the current situation and the sets of previously obtained word-percept (WP) and percept-word (PW) pairs and returns sets of grounded words (GW) and percepts (GP).

procedure GROUNDING(W, P, WP, PW)
  Update WP, and PW using W and P
  for $j = 1$ to $word\_number$ **do**
    Save highest WP to GW
  end for
  for $j = 1$ to $percept\_number$ **do**
    Save highest PW to GP
  end for
  return GW $\cup$ GP
end procedure

is repeated for PW to assign synonymous percepts to the same word[6]. Finally, the sets of GW and GP are merged. Thus, all words are mapped to all corresponding percepts. Algorithm 3 summarizes the grounding procedure.

5 Results and discussion

In several previous studies, RL and CSL have been used for action learning and grounding, respectively (Stulp *et al.,* 2012; Roesler *et al.* 2018). However, *to the best of our knowledge*, there have not been many study investigating simultaneous action learning and grounding, and they significantly differed in their approaches and experimental setups. In this study, the initial and goal positions as well as the corresponding instructions are generated randomly, with the only constraint that they must be valid, e.g. two objects cannot be on the same position. Overall 100,000 different situations have been used. The following sections describe the results for the RL as well as the CSL components.

5.1 Reinforcement learning

The reinforcement learner required for the first 100 situations on average up to 50 episodes until it converged to the optimal policy, when using a continuously decreasing exploration rate that is shared across situations (Figure 1). After around 1,300 situations, the average number of required episodes was below 10 and after about 60,000 situations it was down to one episode, which means that it learned the optimal

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

https://doi.org/10.1017/S0269888919000079 Published online by Cambridge University Press

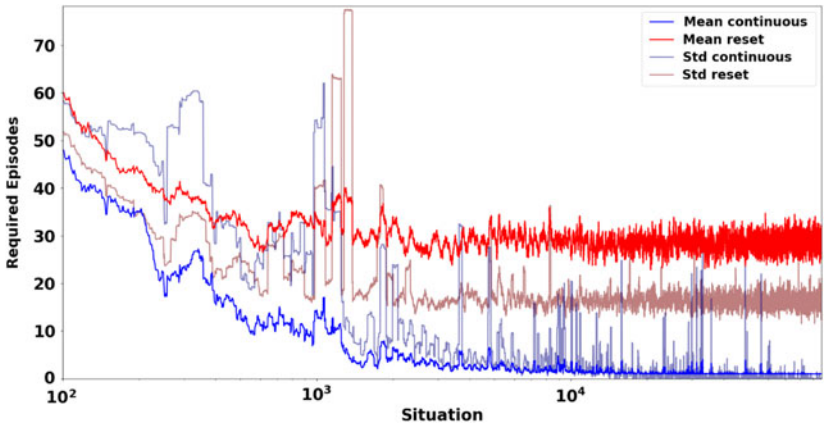

Figure 1 Average number of required episodes until the RL algorithm converges to the optimal policy for random initial positions. The blue curves show the rolling mean and standard deviation, when using a continuously decreasing exploration rate, while the red curves show the rolling mean and standard deviation, when resetting the exploration rate for each situation

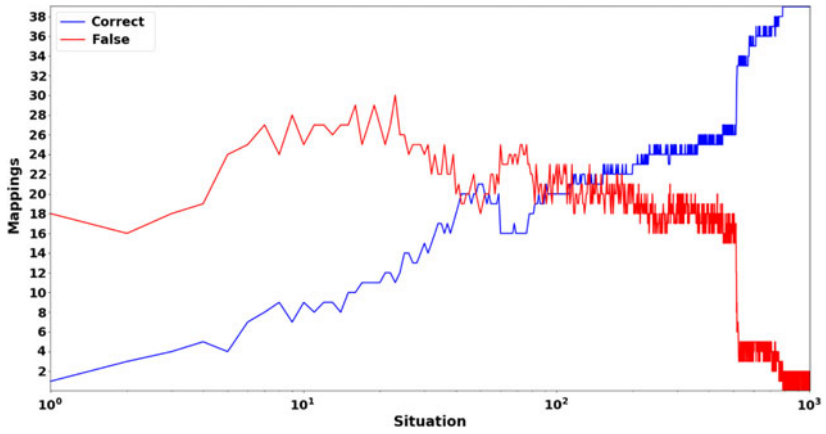

Figure 2 Cross-situational learning results for random situations. The number of correct and false mappings is shown in blue and red, respectively. The figure shows only the first 1000 situations, because afterward all words have been successfully grounded

policies for nearly all situations. When the exploration rate was reset for each situation, the reinforcement learner required around 60 episodes during the first 100 situations. After about 9,000 situations, the average number of required episodes is down to 28. That the agent did not execute the optimal policy immediately, is due to the high exploration rate at the beginning of each situation because it was reset. The results show that a continuously decreasing exploration rate that is shared across situations works best for the investigated scenario; however, it is not clear whether a continuously decreasing exploration rate might cause problems, if a new situation occurs the first time after several million or even billion of situations.

5.2 Cross-situational learning

The employed cross-situational learning algorithm is able to successfully ground all 39 words used in this study through their corresponding percepts. During the first situations, most created mappings are false because the algorithm has not much data available. After around 40 situations, the number of correct mappings equals for the first time the number of false mappings (Figure 2). Interestingly, which words are correctly grounded changes frequently, that is, even if the number of overall correctly grounded words increases, the algorithm might start to ground a word incorrectly after grounding it correctly for many situations. Figure 3 shows that especially action words, i.e. verbs, which are shown

https://doi.org/10.1017/S0269888919000079 Published online by Cambridge University Press

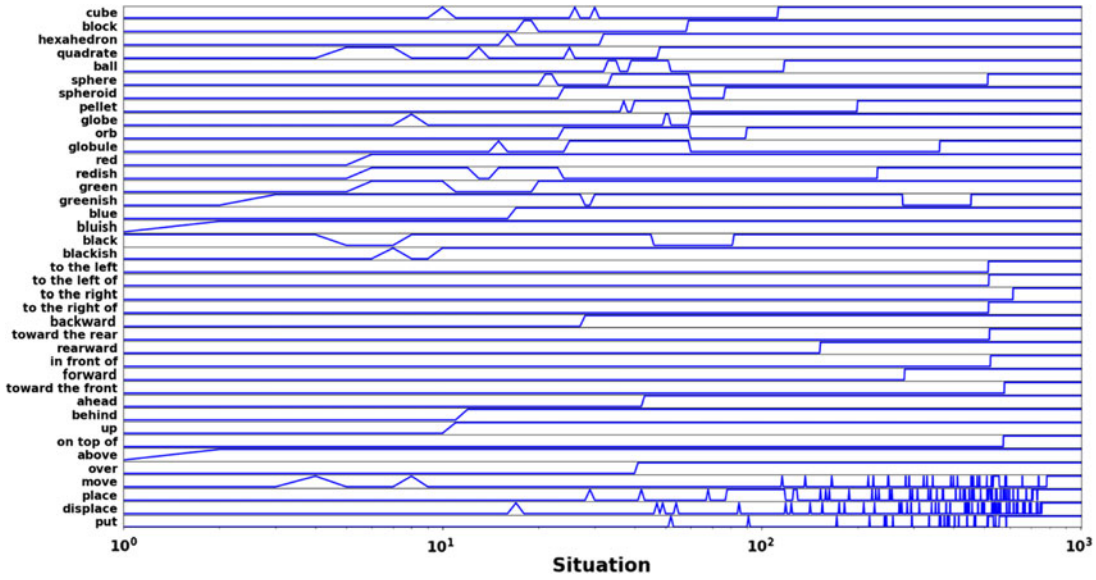

Figure 3 Illustration of correct word mappings for all words used in this study. The figure shows only the first 1000 situations because afterward all words have been successfully grounded

at the bottom of the figure, switch frequently from correctly to incorrectly grounded until they finally are correctly grounded permanently. Overall, it takes nearly 800 situations until all words are grounded successfully. Afterward, the number of correct mappings is constantly 39, while the number of false mappings oscillates between 0 and 2 (Figure 2). This is possible because the algorithm allows a word to be grounded through several percepts, even though there is always only one correct percept for all words in this study. The two additional incorrect mappings are for different word combinations, i.e. there is not a specific incorrect word-percept pair that is temporarily created, but several different word-percept pairs depending on the most recently encountered situations.

6 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 and phrases 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 and phrases, including synonyms, during a simulated human-robot interaction. It also successfully identified auxiliary words and phrases through cross-situational learning. However, 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 investigate grounding of several macro-actions, which will require the creation of several Q-tables. Finally, we will introduce synonymous percepts to examine whether the model is able to create the correct mappings.

References

Abdo, N., Spinello, L., Burgard, W. & Stachniss, C. 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.

Akhtar, N. & Montague, L. 1999. Early lexical acquisition: the role of cross-situational learning. *First Language* **19**(57), 347–358.

https://doi.org/10.1017/S0269888919000079 Published online by Cambridge University Press

Aly, A., Taniguchi, A. & Taniguchi, T. 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, September 2017.

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

Blythe, R. A., Smith, K. & Smith, A. D. M. 2010. Learning times for large lexicons through cross-situational learning. *Cognitive Science* **34**, 620–642.

Carey, S. 1978. The child as word-learner. In *Linguistic Theory and Psychological Reality,* Halle, M., Bresnan, J. & Miller, G. A. (eds). MIT Press, 265–293.

Carey, S. & Bartlett, E. 1978. Acquiring a single new word. *Papers and Reports on Child Language Development* **15**, 17–29.

Dawson, C. R., Wright, J., Rebguns, A., Escárcega, M. V., Fried, D. & Cohen, P. R. 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, August 2013.

Fisher, C., Hall, D. G., Rakowitz, S. & Gleitman, L. 1994. When it is better to receive than to give: syntactic and conceptual constraints on vocabulary growth. *Lingua* **92**, 333–375.

Flanagan, R., Bowman, M. C. & Johansson, R. S. 2006. Control strategies in object manipulation tasks. *Current Opinion in Neurobiology* **16**, 650–659.

Fontanari, J. F., Tikhanoff, V., Cangelosi, A., Ilin, R. & Perlovsky, L. I. 2009a. Cross-situational learning of object-word mapping using neural modeling fields. *Neural Networks* **22**(5–6), 579–585.

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

Gillette, J., Gleitman, H., Gleitman, L. & Lederer, A. 1999. Human simulations of vocabulary learning. *Cognition* **73**, 135–176.

Gu, S., Holly, E., Lillicrap, T. & Levine, S. 2017. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, May–June 2017.

Gudimella, A., Story, R., Shaker, M., Kong, R., Brown, M., Shnayder, V. & Campos, M. 2017. Deep reinforcement learning for dexterous manipulation with concept networks. *CoRR*. https://arxiv.org/abs/1709.06977.

Harnad, S. 1990. The symbol grounding problem. *Physica D* **42**, 335–346.

International Federation of Robotics. 2017. World robotics 2017 - service robots.

Kemp, C. C., Edsinger, A. & Torres-Jara, E. 2007. Challenges for robot manipulation in human environments. *IEEE Robotics & Automation Magazine* **14**(1), 20–29.

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

Pinker, S. 1989. *Learnability and Cognition*. MIT Press.

Popov, I., Heess, N., Lillicrap, T., Hafner, R., Barth-Maron, G., Vecerik, M., Lampe, T., Tassa, Y., Erez, T. & Riedmiller, M. 2017. Data-efficient deep reinforcement learning for dexterous manipulation. *CoRR*. https://arxiv.org/abs/1704.03073.

Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, Inc.

Roesler, O., Aly, A., Taniguchi, T. & Hayashi, Y. 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, May 2018.

Roesler, O., Aly, A., Taniguchi, T. & Hayashi, Y. 2019. Evaluation of word representations in grounding natural language instructions through computational human–robot interaction. In *Proceedings of the 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Daegu, South Korea, March 2019.

Rusu, R. B., Bradski, G., Thibaux, R. & Hsu, J. 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, October 2010, 2155–2162.

She, L., Yang, S., Cheng, Y., Jia, Y., Chai, J. Y. & Xi, N. 2014. Back to the blocks world: learning new actions through situated human-robot dialogue. In *Proceedings of the SIGDIAL 2014 Conference*, Philadelphia, USA, June 2014, 89–97.

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

Smith, A. D. M., & Smith, K. 2012. *Cross-Situational Learning*. Springer US, 864–866. ISBN 978-1-4419-1428-6. doi: 10.1007/978-1-4419-1428-6_1712. https://doi.org/10.1007/978-1-4419-1428-6_1712.

Smith, K., Smith, A. D. M. & Blythe, R. A. 2011. Cross-situational learning: an experimental study of word-learning mechanisms. *Cognitive Science* **35**(3), 480–498.

Smith, L. & Yu, C. 2008. Infants rapidly learn word-referent mappings via cross-situational statistics. *Cognition* **106**, 1558–1568.

Steels, L. & Loetzsch, M. 2012. The grounded naming game. In *Experiments in Cultural Language Evolution*, Steels, L. (ed). John Benjamins, 41–59.

Stulp, F., Theodorou, E. A. & Schaal, S. 2012. Reinforcement learning with sequences of motion primitives for robust manipulation. *IEEE Transactions on Robotics (T-RO)* **28**(6), 1360–1370.

Sutton, R. S. & Barto, A. G. 1998. *Reinforcement Learning: An Introduction*. MIT Press.

Taniguchi, A., Taniguchi, T. & Cangelosi, A. 2017. Cross-situational learning with Bayesian generative models for multimodal category and word learning in robots. *Frontiers in Neurorobotics* **11**.

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

Vogt, P. 2012. Exploring the robustness of cross-situational learning under Zipfian distributions. *Cognitive Science* **36**(4), 726–739.