

Unsupervised Online Grounding of Natural Language during Human-Robot Interactions

Oliver Roesler

Artificial Intelligence Lab

Vrije Universiteit Brussel

Brussels, Belgium

oliver@roesler.co.uk

Abstract

Allowing humans to communicate through natural language with robots requires connections between words and percepts. The process of creating these connections is called symbol grounding and has been studied for nearly three decades. Although many studies have been conducted, not many considered grounding of synonyms and the employed algorithms either work only offline or in a supervised manner. In this paper, a cross-situational learning based grounding framework is proposed that allows grounding of words and phrases through corresponding percepts without human supervision and online, i.e. it does not require any explicit training phase, but instead updates the obtained mappings for every new encountered situation. The proposed framework is evaluated through an interaction experiment between a human tutor and a robot, and compared to an existing unsupervised grounding framework. The results show that the proposed framework is able to ground words through their corresponding percepts online and in an unsupervised manner, while outperforming the baseline framework.

1 Introduction

An increasing number of service robots is employed in human-centered complex environments and interacts with humans on a regular basis. This creates a need for robots that are able to understand instructions provided in natural language, such as *bring a glass of water* or *pick up a box*, to execute them appropriately and thereby enable efficient collaboration with humans. To this end, connections between words, i.e. abstract symbols, and their corresponding percepts, i.e. meanings, need to be created because according to the "Symbol Grounding Problem", which was proposed in 1990 by Har- nad (1990), abstract knowledge and language only

has meaning, if it is linked to the physical world through mappings from words to corresponding percepts.

Grounding approaches can in general be separated into supervised and unsupervised approaches. The former utilize the guidance of a human tutor, while the latter do not require any supervision and try to use co-occurrence information to identify through which percepts a word is grounded. Previous studies, such as (Kollar et al., 2010; Tellex et al., 2011; Aly and Taniguchi, 2018), that investigated unsupervised grounding employed algorithms that only work offline, i.e. these algorithms need to be trained before deployment with in advance collected perceptual data and words, which prevents these algorithms from being used in real-time human-robot interactions. Additionally, most previous studies did not consider ambiguous words, although the sentences humans produce are often ambiguous due to homonymy, i.e. one word refers to several percepts, and synonymy, i.e. one percept can be referred to by several different words. The latter do not need to be true synonyms, i.e. words that refer to the exact same meaning, instead, words only need to be synonyms as references to a percept in a particular set of situations, e.g. *coca cola* or *lemonade* instead of *bottle*.

In this paper, a recently proposed unsupervised online grounding framework (Roesler and Nowé, 2019) is extended to handle real percepts obtained during human-robot interactions. More specifically, the learning framework is extended to first convert obtained percepts through clustering to an abstract representation, which is then used to ground all non-auxiliary words¹ of the encountered natural language instructions through cross-situational learning. Each shape, color, and action is referred to by at least two synonymous words, which need to

¹Auxiliary words are words that do not have corresponding percepts and only exist for grammatical reasons.

be mapped to their corresponding geometric characteristics, color histograms, or kinematic features of the robot joints during action execution, to investigate the ability of the used frameworks to handle synonymous words. The grounding performance of the proposed framework is evaluated by comparing it to the grounding performance of a Bayesian grounding framework that has been used in several previous studies, e.g. (Aly and Taniguchi, 2018; Roesler et al., 2018, 2019)

The rest of this paper is structured as follows: Sections (2 and 3) provide a brief overview of cross-situational learning and related work. Afterwards, an overview of the proposed unsupervised online grounding framework as well as the unsupervised Bayesian baseline framework is given in Sections (4 and 5). The experimental design and obtained results are described in Sections (6 and 7). Finally, Section (8) concludes the paper.

2 Background

Cross-situational learning (CSL) refers to the process of learning the meaning of words across multiple exposures to handle referential uncertainty. The basic idea is that a set of candidate meanings, i.e. mappings from words to percepts, can be created for every situation or context a word is used in and that the correct meaning can be obtained by determining the intersection of the sets of candidate meanings (Pinker, 1989; Fisher et al., 1994). Thus, the correct mapping between a word and its corresponding percepts, i.e. its meaning, will reliably reoccur across situations (Blythe et al., 2010; Smith and Smith, 2012). A number of experimental studies have confirmed that humans use CSL for word learning, when no prior knowledge of language is available (Akhtar and Montague, 1999; Gillette et al., 1999; Smith and Yu., 2008). Since CSL requires more than one exposure to learn a word, it belongs to the group of slow-mapping mechanisms through which most words are acquired (Carey, 1978). In contrast, *fast-mapping* allows words to be acquired through a single exposure, but it is only used for a limited number of words and can neither be explained nor achieved through CSL (Carey and 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).

3 Related Work

Grounding is used to obtain the meaning of an abstract symbol, e.g. a word, by linking it to perceptual information, i.e. the “real” world (Harnad, 1990). There exist many different approaches for grounding. She et al. (2014) grounded higher level symbols through already grounded lower level symbols with the help of a dialog system. Since the system requires a sufficiently large set of grounded lower level symbols as well as a professional tutor to answer its questions, its usefulness is limited. The need for a human tutor that knows the correct mappings also limits the applicability of the *Naming Game*, which allows an agent to quickly learn word-percept mappings (Steels and Loetzsch, 2012). In contrast to the previous approaches, cross-situational learning (Section 2), which assumes that one word appears several times together with the same perceptual feature vector so that a corresponding mapping can be created, does not require a human tutor for grounding (Siskind, 1996; Smith et al., 2011). Previous studies investigated the use of cross-situational learning for grounding of objects, actions, and spatial concepts (Roesler et al., 2019; Dawson et al., 2013). In most studies, grounding was conducted offline, i.e. perceptual data and words were collected in advance, which prevents these approaches from being used in real-time human-robot interactions. In contrast to these approaches, the framework used in this study learns the correct mappings from words to percepts online while interacting with humans and does not require separate training and test phases. Additionally, the majority of employed models were not able to handle ambiguous words, although, the sentences humans produce are often ambiguous due to homonymy and synonymy. One recent study showed that grounding of known 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 (Roesler et al., 2018). Therefore, the online grounding mechanism employed in this study uses no additional semantic or syntactic information to ground synonyms.

4 Grounding Framework

The employed grounding framework consists of four parts: (1) 3D object segmentation component, which segments objects into point clouds to determine their geometric characteristics and colors, (2)

Action recording component, which creates action feature vectors by recording the states of several joints while the robot is executing actions, (3) Percept clustering component, which obtains an abstract representation of percepts through clustering, and (4) Cross-situational learning based grounding component, which identifies auxiliary words 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.

1. 3D object segmentation:

- **Input:** Point cloud data.
- **Output:** Geometric characteristics and colors of objects.

2. Action recording:

- **Input:** Changes of the robot’s joint states during action execution.
- **Output:** Action feature vectors representing the executed actions.

3. Clustering of percepts:

- **Input:** Geometric object characteristics, object colors, and action feature vectors.
- **Output:** Cluster numbers of percepts.

4. Cross-situational learning:

- **Input:** Natural language instructions and cluster numbers of percepts.
- **Output:** Word to percept mappings.

4.1 3D Object Features

In this study, an unsupervised model based 3D point cloud segmentation approach is used to segment objects lying in a plane into separate point clouds because it is fast, reliable and does not need much prior knowledge, such as object models or the number of regions to process (Craye et al., 2016). The applied model uses the RANSAC algorithm (Fischler and Bolles, 1981) to detect the major plane in the environment, which is a tabletop in the conducted experiment, and keeps track of it in consecutive frames. If a plane is orthogonal to the major plane and touches at least one border of the image, it is defined as a wall plane. After filtering out points that belong to the main plane or wall planes, the remaining points are voxelized and clustered into blobs representing object candidates. Blobs that are neither extremely small nor large are

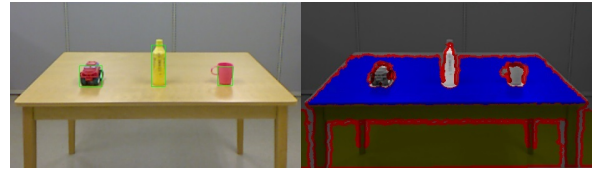


Figure 1: Illustration of the used objects and the corresponding 3D point cloud information: (A) car, (B) bottle, and (C) cup.

treated as objects². Point clouds of segmented objects are characterized through Viewpoint Feature Histogram (VFH) descriptors (Rusu et al., 2010), which represent the object geometries taking into consideration the viewpoints while ignoring scale variances, and color histograms, which represent the colors of the objects. Figure (1) provides an illustrative example of the obtained 3D point cloud information.

4.2 Action Features

Action feature vectors are used to represent the dynamic characteristics of actions during execution through teleoperation. Overall, five different characteristics, which represent possible subactions, are recorded through the sensors of the robot (Toyota Motor Corporation, 2017). The used characteristics are:

1. The distance from the actual to the lowest torso position in meters.
2. The angle of the arm flex joint in radians.
3. The angle of the wrist roll joint in radians.
4. Velocity of the base.
5. Binary state of the gripper (1: closing, 0: opening or no change).

They are then combined into the following vector:

$$\begin{pmatrix} a_1^1 & a_1^2 & a_1^3 & a_1^4 & a_1^5 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_6^1 & a_6^2 & a_6^3 & a_6^4 & a_6^5 \end{pmatrix},$$

where a^1 represents the difference of the distances from the lowest torso position in meters, while a^2 and a^3 represent the differences in the angles of the arm flex and wrist roll joints in radians, respectively. The differences are calculated by subtracting the

²The threshold for the blob size was manually set based on the objects used in the experiment and should be suitable for all objects of similar size.

values at the beginning of the subaction from the values at the end of the subaction. a^4 represents the mean velocity of the base (forward/backward), and a^5 represents the binary gripper state. Each action is characterized through six manually defined subactions. Therefore, if an action consists of less than six subactions, rows with zeros are added at the end, while the duration of a subaction is not fixed because it depends on the teleoperator.

4.3 Clustering of percepts

The CSL algorithm (Section 4.4) requires percepts to be converted to an abstract representation that can then be used to ground natural language. The abstract representation is obtained through clustering as proposed in (Roesler, 2019). Since it cannot be assumed that the number of clusters, i.e. the number of different percepts, is known in advance, DBSCAN, which is a density-based clustering algorithm proposed by Ester et al. (1996), is used³ because it determines the number of clusters automatically, while only requiring two parameters, i.e. the radius ϵ and threshold $minSamples$. Each iteration DBSCAN determines a number of core points, which are points that have more than $minSamples$ points within radius ϵ around them (Schubert et al., 2017). All the points within radius ϵ of a core point are assigned to the same cluster as the core point. Cluster numbers are calculated every situation prior to grounding so that they can be provided to the CSL algorithm. Recalculating them every situation is necessary to take into account the new percepts of that situation.

4.4 Cross-Situational Learning

A variety of algorithms have been developed that realize CSL in different ways, e.g. through the use of probabilistic models (Aly and Taniguchi, 2018; Roesler et al., 2019), to ground words through percepts in artificial agents. This section describes an online CSL algorithm for grounding of words, which has first been proposed by Roesler and Nowé (2018) and recently been extended with auxiliary word and phrase detection (Roesler and Nowé, 2019). Since the sentences in this study are shorter, have a much simpler structure, and less variation than the sentences used in (Roesler and Nowé, 2019), the previous auxiliary word and phrase detection algorithms do not work.

³The used DBSCAN implementation is available in *scikit-learn* (Pedregosa et al., 2011).

Algorithm 1 The grounding procedure takes as input all words (W) and percepts (P) of the current situation, the sets of all previously obtained word-percept (WP) and percept-word (PW) pairs, the set of auxiliary words (AW), and the set of permanent phrases (PP) and returns the sets of grounded words (GW) and percepts (GP).

```

1: procedure GROUNDING( $W, P, WP, PW, AW, PP$ )
2:   Substitute words with phrases from  $PP$ 
3:   Update  $AW$  (Algorithm 2) and remove  $AW$  from  $W$ 
4:   Update  $WP$  and  $PW$  using  $W$  and  $P$ 
5:   for  $j = 1$  to  $word\_number$  do
6:     Save highest  $WP$  to  $GW$ 
7:   end for
8:   for  $j = 1$  to  $percept\_number$  do
9:     Save highest  $PW$  to  $GP$ 
10:  end for
11:  return  $GW \cup GP$ 
12: end procedure

```

Thus, a novel auxiliary word detection algorithm (Algorithm 2) is proposed to handle the simpler sentences employed in this study⁴, while no phrase detection is used to ensure a fair comparison with the baseline framework (Section 5), which does not have any phrase detection capabilities. The rest of this section provides an overview of the employed grounding algorithm.

For each situation all corresponding words and percepts are given to the grounding algorithm (Algorithm 1), while the sets of grounded words (GW) and percepts (GP) are initially empty. Before the actual grounding procedure, words that are part of known phrases will be combined so that they can be grounded together and auxiliary words are automatically detected and removed (Algorithm 2). Afterwards, all possible word-percept (WP) and percept-word (PW) pairs are created, i.e. for each word and percept a set containing all percepts and words they occurred with is created, and saved together with a number indicating how often the pair occurred. The highest word-percept pair is determined and saved to the set of grounded words (GW). All other word-percept pairs the word or

⁴Both auxiliary word mechanisms, i.e. the one used in (Roesler and Nowé, 2019) and the one proposed in this study, are used in parallel because both have shown to not produce false detections, i.e. they either detect an auxiliary word correctly or do not detect it.

Algorithm 2 The auxiliary word detection procedure takes as input the sets of word and percept occurrences (WO and PO), and the set of detected auxiliary words (AW).

```

1: procedure AUXILIARY WORD DETECTION( $WO, PO, AW$ )
2:   for word, occurrence in  $WO$  do
3:     if  $occurrence > \max(PO) * 2$  then
4:       Add word to  $AW$ 
5:     end if
6:   end for
7:   return  $AW$ 
8: end procedure

```

percept are part of will no longer be considered for the selection of the highest word-percept pair in future iterations. This restriction is applied until all percepts have been used once for grounding. Afterwards, if some words have not been grounded, all percepts will become again available for grounding until all words have been grounded to allow grounding of synonyms. After all words have been grounded the same process is repeated for percept-word pairs to assign synonymous percepts to the same word. Finally, the sets of grounded words and percepts are merged.

5 Baseline Framework

The baseline framework consists of three parts: (1) 3D object segmentation component as described in Section (4.1), (2) Action recording component as described in Section (4.2), and (3) Bayesian learning model, which identifies auxiliary words and grounds non-auxiliary words and phrases through corresponding percepts. Since the perceptual data extraction components are the same for both frameworks, any difference in grounding performance can only be due to the different grounding algorithms, i.e. component three and four of the proposed framework (Sections 4.3 and 4.4) and component three of the baseline framework, which is described in the remainder of this section.

The probabilistic learning model, described in this section, is based on the model used in (Roesler et al., 2019), since the experimental setup employed in this study (Section 6) is also based on the scenario used in (Roesler et al., 2019). In general, the model has been chosen as a baseline because similar models have previously been em-

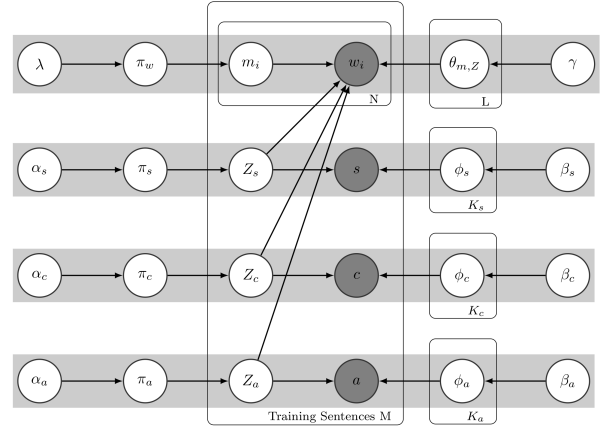


Figure 2: Graphical representation of the probabilistic model. Indices \mathbf{i} , \mathbf{s} , \mathbf{c} , and \mathbf{a} denote the order of words, object shapes, object colors, and actions, respectively.

Table 1: Definitions of the learning parameters in the graphical model.

Parameter	Definition
λ	Hyperparameter of the distribution π_w
$\alpha_s, \alpha_c, \alpha_a$	Hyperparameters of the distributions π_s, π_c and π_a
m_i	Modality index of each word (modality index $\in \{\text{Shape, Color, Action, AW}\}$)
Z_s, Z_c, Z_a	Indices of shape, color and action distributions
w_i	Word indices
s, c, a	Observed states representing shapes, colors and actions
γ	Hyperparameter of the distribution $\theta_{m,Z}$
$\beta_s, \beta_c, \beta_a$	Hyperparameters of the distributions ϕ_s, ϕ_c and ϕ_a
$\theta_{m,Z}$	Word distribution over modalities

ployed in similar grounding scenarios by different researchers, e.g. (Kollar et al., 2010; Tellex et al., 2011; Aly and Taniguchi, 2018; Roesler et al., 2018, 2019). In the model, the observed state w_i represents word indices, i.e. each individual word is represented by a different integer⁵. The observed state s represents the shape of objects, more specifically their geometric characteristics expressed through VFH descriptors (Section 4.1), c represents the color of objects and a represents actions. Table (1) provides a summary of the definitions of the learning model parameters. The corresponding probability distributions, i.e., $w_i, \theta_{m,Z_{L_1}}, \phi_{s_{K_1}}, \phi_{c_{K_2}}, \phi_{a_{K_3}}, \pi_w, \pi_s, \pi_c, \pi_a, m_i, Z_s, Z_c, Z_a, s, c,$ and a , which characterize the different modalities in the graphical model, are defined in Equation (1), where Cat denotes a categorical distribution, Dir denotes a Dirichlet distribution, GIW denotes a Gaussian Inverse-Wishart distribution, and N denotes a mul-

⁵The following two example sentences illustrate the representation of words through word indices: (please, **1**) (lift up, **2**) (the, **3**) (brown, **4**) (coke, **5**) and (lift up, **2**) (the, **3**) (brownish, **6**) (lemonade, **7**), where the bold numbers indicate word indices.

tivariate Gaussian distribution.

$$\left\{ \begin{array}{l} w_i \sim \text{Cat}(\theta_{m_i, Z_{m_i}}) \\ \theta_{m, Z_{L_1}} \sim \text{Dir}(\gamma) \quad , \quad L_1 = (1, \dots, L) \\ \phi_{s_{K_1}} \sim \text{GIW}(\beta_s) \quad , \quad K_1 = (1, \dots, K_s) \\ \phi_{c_{K_2}} \sim \text{GIW}(\beta_c) \quad , \quad K_2 = (1, \dots, K_c) \\ \phi_{a_{K_3}} \sim \text{GIW}(\beta_a) \quad , \quad K_3 = (1, \dots, K_a) \\ \pi_w \sim \text{Dir}(\lambda) \\ \pi_s \sim \text{Dir}(\alpha_s) \\ \pi_c \sim \text{Dir}(\alpha_c) \\ \pi_a \sim \text{Dir}(\alpha_a) \\ m_i \sim \text{Cat}(\pi_w) \\ Z_s \sim \text{Cat}(\pi_s) \\ Z_c \sim \text{Cat}(\pi_c) \\ Z_a \sim \text{Cat}(\pi_a) \\ s \sim N(\phi_{Z_s}) \\ c \sim N(\phi_{Z_c}) \\ a \sim N(\phi_{Z_a}) \end{array} \right. \quad (1)$$

The latent variables of the Bayesian learning model are inferred using the Gibbs sampling algorithm (Geman and Geman, 1984) (Algorithm 3), which repeatedly samples from and updates the posterior distributions (Equation 2). Distributions were sampled for 100 iterations, after which convergence had been achieved.

$$\left\{ \begin{array}{l} \phi_s \sim P(\phi_s | s, \beta_s) \\ \phi_c \sim P(\phi_c | c, \beta_c) \\ \phi_a \sim P(\phi_a | a, \beta_a) \\ \pi_w \sim P(\pi_w | \lambda, m) \\ \pi_s \sim P(\pi_s | \alpha_s, Z_s) \\ \pi_c \sim P(\pi_c | \alpha_c, Z_c) \\ \pi_a \sim P(\pi_a | \alpha_a, Z_a) \\ Z_s \sim P(Z_s | s, \pi_s, w) \\ Z_c \sim P(Z_c | c, \pi_c, w) \\ Z_a \sim P(Z_a | a, \pi_a, w) \\ \theta_{m, Z} \sim P(\theta_{m, Z} | m, Z_s, Z_c, Z_a, \gamma, w) \\ m_i \sim P(m_i | \theta_{m, Z}, Z_s, Z_c, Z_a, \pi_w, w_i) \end{array} \right. \quad (2)$$

6 Experimental Setup

The experimental scenario used in this study is based on the scenario used in (Roesler et al., 2019). The main difference is the use of an additional modality, i.e. color, which leads to slightly different sentences. During the experiment a human tutor and HSR robot⁶ interact in front

⁶The Human Support Robot from Toyota, which is used for the experiment, has an omnidirectional movable cylindrical

Algorithm 3 Inference of the model’s latent variables. In this study, *nr_of_iterations* was set to 100.

```

1: procedure GIBBS SAMPLING( $W, P, WP, AW$ )
2:   Initialization of  $\theta, \phi_s, \phi_c, \phi_a, \pi_w, \pi_s, \pi_c,$ 
3:    $\pi_a, Z_s, Z_c, Z_a, m_i$ 
4:   for  $i = 1$  to nr_of_iterations do
5:     Equation (2)
6:   end for
7:   return  $\theta, \phi_s, \phi_c, \phi_a, \pi_w, \pi_s, \pi_c, \pi_a, Z_s,$ 
8:      $Z_c, Z_a, m_i$ 
9: end procedure

```

of a table, with one of the following five objects {BOTTLE, CUP, BOX, CAR, and BOOK} (Figure 1). Each interaction follows below procedure:

1. The human tutor places an object on the table and the robot determines the object’s geometric characteristics and color to create corresponding feature vectors (Section 4.1).
2. An instruction, which describes how to manipulate the object, is given to the robot by the human tutor, e.g. “please lift up the red soda”.
3. The human tutor teleoperates the robot to execute the action provided through the instruction while several kinematic characteristics are recorded and converted into an action feature vector (Section 4.2).

A total of 125 interactions were performed to record perceptual information for all combinations of employed shapes, colors, and actions. Since instruction words were selected randomly for each situation, except that words had to fit the encountered percepts, their number of occurrences in the data varies, e.g. the word “coffee” only occurs once, while the word “brown” occurs 14 times. Grounding was then performed for ten different interaction sequences, i.e. the order of the recorded situations was randomly changed, to ensure that the performance is not due to the specific order in which situations are encountered. Figure (3) shows how often each word occurred on average in all interactions as well as the training and test interactions.

shaped body with one arm and gripper. It is equipped with a variety of different sensors, such as stereo and wide-angle cameras, and has 11 degrees of freedom.

Table 2: Overview of all percepts with their corresponding synonyms. The action percepts are explained in Table (3).

Type	Percept	Synonyms
Shape	Bottle	coca cola, soda, peps, coke, lemonade
	Cup	latte, milk, milk tea, coffee, espresso
	Box	candy, chocolate, confection, sweets, dark chocolate
	Car	audi, toyota, mercedes, bmw, honda
	Book	harry potter, narnia, lord of the rings, dracula, frankenstein
Color	Yellow	yellow, yellowish
	Pink	pink, pinkish
	Brown	brown, brownish
	Red	red, reddish
	White	white, whitish
Action	Lift up	lift up, raise
	Grab	grab, take
	Push	push, poke
	Pull	pull, drag
	Move	move, shift
Auxiliary Word	-	the
	-	please

Table 3: Explanations of the employed action percepts.

Percept	Description
Lift up	The object will be grabbed and lifted up.
Grab	The object will be grabbed, but not displaced.
Push	The object will be pushed with the closed gripper without being grabbed first.
Pull	The object will be grabbed and moved towards the robot.
Move	The object will be grabbed and moved away from the robot.

Each sentence consists of one of the following structures: “*action the color shape*” or “*please action the color shape*”, where *action*, *color*, and *shape* are substituted by one of their corresponding words (Table 2). Each action and color can be referred to by two different words, while each shape has five corresponding words. During training and testing the obtained situations are given to the proposed and baseline frameworks. The former framework gets the situations separately one after the other, as if it is processing the data in real-time during the interaction. It first clusters the percepts of the current situation together with all previously encountered percepts to obtain abstract representations of shapes, colors and actions (Section 4.3). Afterwards, the CSL based grounding algorithm is used to ground words through the obtained cluster numbers (Section 4.4). In contrast, the baseline framework does not allow online learning and requires all sentences and corresponding percepts of the training situations to be given at once to the learning model.

7 Results and Discussion

The proposed cross-situational learning framework (Section 4) is evaluated through a human-robot interaction scenario (Section 6) and the obtained grounding results are compared to the groundings achieved by an unsupervised Bayesian grounding framework (Section 5). Figure (4) shows how

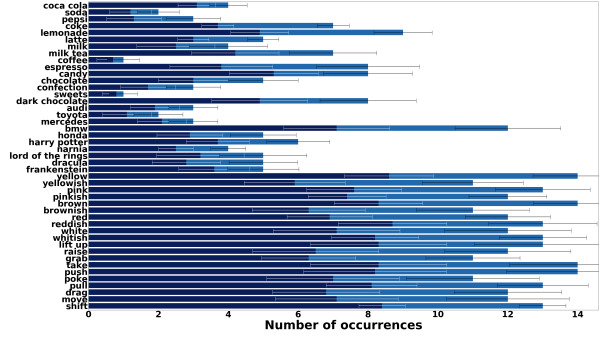


Figure 3: Word occurrences for all encountered words except auxiliary words. The dark blue part of the bars shows the mean number of occurrences during training and the bright blue part the mean number of occurrences during testing.

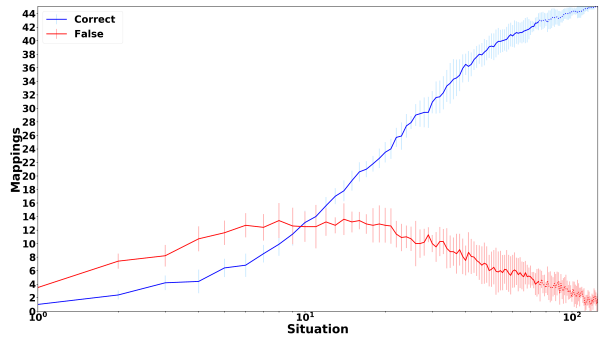
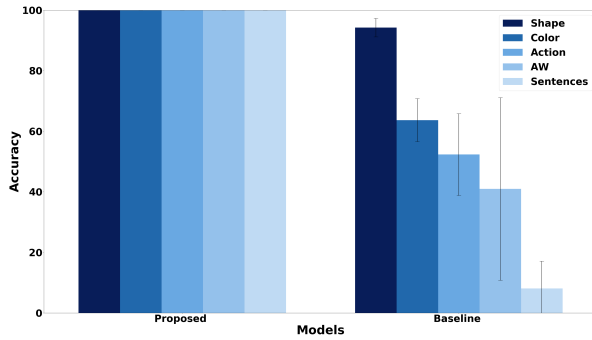
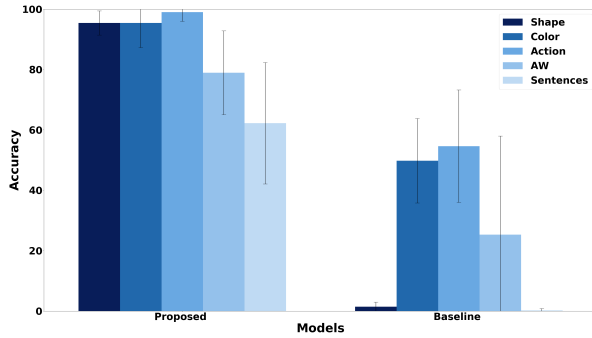


Figure 4: Mean number and standard deviation of correct and false mappings obtained by the proposed model over all 125 situations. The dotted part only occurs, when all situations are used for training, otherwise the model obtains only 43 correct mappings.

the mean number of correct and false mappings changes, when the proposed grounding framework encounters the employed situations one after the other. It also shows that all 45 correct mappings are obtained, when all 125 situations are used for training, while on average only 43 correct mappings are obtained, when only 60% of the situations are used for training. The figure also illustrates the online grounding capability of the model, i.e. that it updates its mappings with every new encountered situation, as well as its transparency because it allows to check at any time through which percept a word is grounded at that moment. Based on the collected co-occurrence information it would also be possible to calculate a confidence score for every mapping to understand how likely it is that a false mapping disappears or a correct mapping persists. The described transparency of the proposed framework can be helpful to understand and debug responses to instructions provided by a human,



(a) All situations used for training and testing.



(b) 60% of the situations used for training and 40% for testing.

Figure 5: Mean grounding accuracy results and corresponding standard deviations for all modalities and both models. Additionally, the percentage of sentences for which all words were correctly grounded is shown.

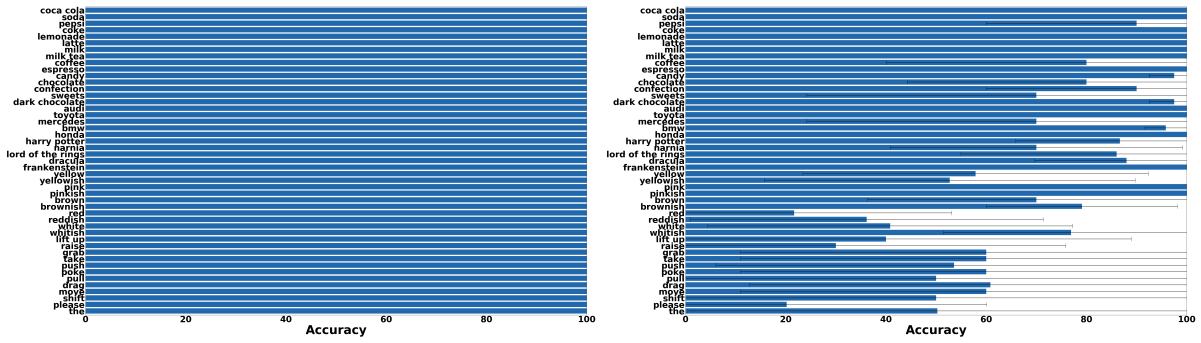
when the framework is used to control an artificial agent interacting with a human, especially when the responses are incorrect or inappropriate.

In contrast, the baseline model requires an explicit training phase so that no corresponding figure, illustrating the number of correct and false mappings, can be created. Thus, to allow a comparison between the two models, the mappings of the proposed model are extracted after 125 and 75 situations, depending on the used train/test split. Two different train/test splits are analyzed in this study. For the first split, all situations are used for training and testing to see how well the frameworks perform when all test situations have been encountered before. For the second split, only 60% of the used situations are provided for training, while the remaining situations are used for testing. In this case, it is possible that some words never occur during training or only a limited number of times, e.g. once or twice. If a word does not occur during training, the proposed model is not able to obtain a corresponding mapping which leads to an accuracy of 0% as shown in Figure (6c) for the words *coffee* and *sweets*, which both only exist once in the dataset and are thus only present during train-

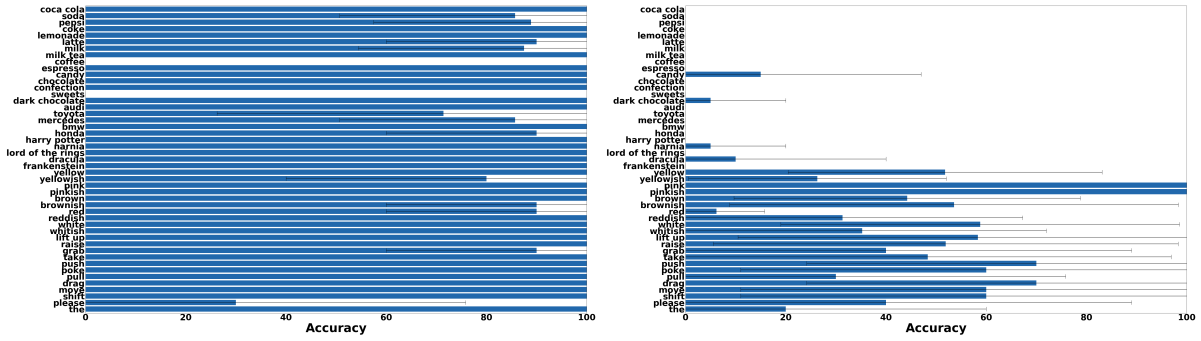
ing or testing, but not both. The word accuracies shown in Figure (6) were calculated by dividing the number of times a word was correctly grounded through the number of times the word was encountered during testing. Similar to the proposed model, the baseline model was also not able to ground the words *coffee* and *sweets* correctly, when only 60% of the situations were used for training. However, the baseline model also seems to require in general a higher minimum number of occurrences to successfully ground words, since there are many words that achieved a mean accuracy of 0%, when only 60% of the situations were used for training (Figure 6d).

Figure (5a) shows that the proposed model achieves perfect grounding, when the same situations are provided for training and testing, which confirms that it is able to obtain all correct mappings as shown in Figure (4). However, if only 60% of the situations are used for training and the remaining 40% unknown situations for testing the grounding accuracy drops for both models. For the proposed model the largest accuracy decrease is seen for auxiliary words, while still more than 95% of the obtained shape, color and action groundings are correct. For the baseline framework the largest drop in accuracy is seen for shapes, from more than 95% to less than 2%. The reason might be that every shape word has 5 synonyms, thus, if words would be equally distributed among all situations and specifically among the training and test sets, the decrease might not be as sharp. However, Figure 3 shows that the number of occurrences is not necessarily the reason for the drop because the words *bmw* and *narnia* occurred on average 7 and 2.5 times during training, respectively, and *narnia* achieved an accuracy of about 5%, while the accuracy of *bmw* was 0% (Figure 6d). In contrast, the proposed model shows a more stable performance, since it was able to ground all non-auxiliary words that occurred at least one time during training with a mean accuracy of more than 70%, while only the auxiliary word *please* achieved a lower mean accuracy of 30%.

Overall the evaluation shows that the proposed model outperforms the baseline model based on its auxiliary word detection and grounding accuracy. Interestingly, the performance difference is larger, when only 60% of the situations are used for training, although this scenario is artificially harming the proposed model by preventing it to learn during



(a) Proposed model using all situations for training and testing. (b) Baseline model using all situations for training and testing.



(c) Proposed model using 60% of the situations for training and 40% for testing. (d) Baseline model using 60% of the situations for training and 40% for testing.

Figure 6: Mean accuracy results and corresponding standard deviations for each individual word.

testing, since it does not require explicit training. In addition to the better grounding performance, the proposed model is also more transparent, which becomes important when robots are interacting with humans in complex and unrestricted environments, especially if some actions of the robots can cause harm to humans.

8 Conclusions and Future Work

This paper investigated a multimodal framework for grounding synonymous shape, color and action words through the visual perception and proprioception of a robot during its interaction with a human tutor. The cross-situational learning model was set up to learn the meaning of shape and color words of objects as well as action words using geometric characteristics and color information of objects obtained from point cloud information as well as kinematic features of the robot joints recorded during action execution.

The proposed model allowed auxiliary word detection and online grounding of synonyms through real percepts in an unsupervised manner and without the use of any syntactic or semantic information. Additionally, it outperformed the baseline model based on the accuracy of the obtained groundings,

its capability to process new situations online and its transparency.

In future work, different mechanisms will be investigated to improve the sample efficiency of the algorithm, which will become relevant, if a larger number of words is used or words occur less often. Additionally, it will be verified whether the framework can handle homonyms. Finally, supervised grounding methods will be integrated so that the robot is able to use human feedback, but does not require it.

References

- N. Akhtar and L. Montague. 1999. Early lexical acquisition: the role of cross-situational learning. *First Language*, 19(57):347–358.
- A. Aly and T. Taniguchi. 2018. Towards understanding object-directed actions: A generative model for grounding syntactic categories of speech through visual perception. In *IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia.
- R. A. Blythe, K. Smith, and A. D. M. Smith. 2010. Learning times for large lexicons through cross-situational learning. *Cognitive Science*, 34:620–642.

- S. Carey. 1978. The child as word-learner. In M. Halle, J. Bresnan, and G. A. Miller, editors, *Linguistic theory and psychological reality*, pages 265–293. MIT Press, Cambridge, MA.
- S. Carey and E. Bartlett. 1978. Acquiring a single new word. *Papers and Reports on Child Language Development*, 15:17–29.
- C. Craye, D. Filliat, and J.-F. Goudou. 2016. Environment exploration for object-based visual saliency learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden.
- 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.
- M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 226–231, Portland, Oregon, USA.
- M. A. Fischler and R. C. Bolles. 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM (CACM)*, 24(6):381–395.
- C. Fisher, D. G. Hall, S. Rakowitz, and L. Gleitman. 1994. When it is better to receive than to give: Syntactic and conceptual constraints on vocabulary growth. *Lingua*, 92:333–375.
- S. Geman and D. Geman. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 6(6):721–741.
- J. Gillette, H. Gleitman, L. Gleitman, and A. Lederer. 1999. Human simulations of vocabulary learning. *Cognition*, 73:135–176.
- S. Harnad. 1990. The symbol grounding problem. *Physica D*, 42:335–346.
- T. Kollar, S. Tellex, D. Roy, and N. Roy. 2010. Toward understanding natural language directions. In *Proceedings of the 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Osaka, Japan.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.
- S. Pinker. 1989. *Learnability and cognition*. MIT Press, Cambridge, MA.
- O. Roesler. 2019. A cross-situational learning based framework for grounding of synonyms in human-robot interactions. In *Proceedings of the Fourth Iberian Robotics Conference (ROBOT)*, Porto, Portugal.
- 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.
- O. Roesler, A. Aly, T. Taniguchi, and Y. Hayashi. 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.
- O. Roesler and A. Nowé. 2018. Simultaneous action learning and grounding through reinforcement and cross-situational learning. In *ALA 2018, Adaptive Learning Agents Workshop.*, Stockholm, Sweden.
- O. Roesler and A. Nowé. 2019. Action learning and grounding in simulated human robot interactions. *The Knowledge Engineering Review*, 34(E13).
- 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)*, pages 2155–2162, Taipei, Taiwan.
- E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu. 2017. DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS)*, 42(3).
- L. She, S. Yang, Y. Cheng, Y. Jia, J. Y. Chai, and N. Xi. 2014. Back to the blocks world: Learning new actions through situated human-robot dialogue. In *Proceedings of the SIGDIAL 2014 Conference*, pages 89–97, Philadelphia, U.S.A.
- J. M. Siskind. 1996. A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition*, 61:39–91.
- A. D. M. Smith and K. Smith. 2012. *Cross-Situational Learning*, pages 864–866. Springer US, Boston, MA.
- 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):480–498.

- L. Smith and C. Yu. 2008. Infants rapidly learn word-referent mappings via cross-situational statistics. *Cognition*, 106:1558–1568.
- L. Steels and M. Loetzsch. 2012. The grounded naming game. In L. Steels, editor, *Experiments in Cultural Language Evolution*, pages 41–59. John Benjamins, Amsterdam.
- 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):6476.
- Toyota Motor Corporation. 2017. *HSR Manual*, 2017.4.17 edition.
- P. Vogt. 2012. Exploring the robustness of cross-situational learning under zipfian distributions. *Cognitive Science*, 36(4):726–739.