Evaluation of Word Representations in Grounding Natural Language Instructions through Computational Human-Robot Interaction

Oliver Roesler, Amir Aly, Tadahiro Taniguchi, Yoshikatsu Hayashi

Evaluation of Word Representations in Grounding Natural Language Instructions through Computational Human-Robot Interaction

Oliver Roesler
Biomedical Engineering
University of Reading
Reading, UK
oliver@roesler.co.uk

Amir Aly
Emergent Systems Laboratory
Ritsumeikan University
Kusatsu, Japan
amir.aly@em.ci.ritsumei.ac.jp

Tadahiro Taniguchi
Emergent Systems Laboratory
Ritsumeikan University
Kusatsu, Japan
taniguchi@em.ci.ritsumei.ac.jp

Yoshikatsu Hayashi
Biomedical Engineering
University of Reading
Reading, UK
y.hayashi@reading.ac.uk

***Abstract*—In order to interact with people in a natural way, a robot must be able to link words to objects and actions. Although previous studies in the literature have investigated grounding, they did not consider grounding of *unknown* synonyms. In this paper, we introduce a probabilistic model for grounding unknown synonymous object and action names using cross-situational learning. The proposed Bayesian learning model uses four different word representations to determine synonymous words. Afterwards, they are grounded through geometric characteristics of objects and kinematic features of the robot joints during action execution. The proposed model is evaluated through an interaction experiment between a human tutor and HSR robot. The results show that semantic and syntactic information both enable grounding of *unknown* synonyms and that the combination of both achieves the best grounding.**

***Index Terms*—Language grounding; Bayesian learning model; Computational human-robot interaction; Cross-situational learning**

I. Introduction

The number of non-industrial robots that are integrated into peoples everyday life is continuously growing. In 2016, more than 67,000 service robots have been sold worldwide to efficiently collaborate with human users in complex environments [21, 31]. To this end, a robot must be able to converse in natural language and understand the instructions of a user so that it executes the desired action appropriately, such as *pick up a drink* or *grab a box*. To meet this target, the robot has to relate words and sensory data that refer to the same object or action to each other, which defines the "Symbol Grounding" problem that was first described in Harnad [18]. However, humans often use synonymous words, i.e. different names, for the same object or action, which makes one-to-one mappings between words and perceptual information not possible to attain. This can be either due to different words in different regional dialects or the specific context in which the instruction is given. Since the robot should be a natural part of the human environment, it must be able to handle this problem.

In this paper, we address the issue of relating unknown synonyms, i.e. synonyms the model has not encountered during training, to the same action or object. More specifically, we present an unsupervised learning model for sensory-motor coupling using a probabilistic learning model and a robot. The Bayesian learning model employs either syntactic-semantic information encoded in the vector representation of words obtained via Word2Vec, syntactic information encoded in POS tags, or both to determine the corresponding object or action for an unseen synonym of a grounded word. The main question we investigate is whether the simple syntactic-semantic vector space provided by Word2Vec and/or the POS tags are good enough to allow for grounding of *unknown* synonyms and if so, which of them performs better. The three representations are compared to a baseline model that does not have any information about synonyms, i.e. it treats synonyms as separate words.

The rest of this paper is structured as follows: Section (II) discusses related work on grounding, and semantic similarity

Fig. 1: Schematic representation of the human-robot interaction scenario. A robot is placed in front of a table with one object, and a human tutor provides an instruction so that the robot executes the corresponding action.

between words. Section (III) provides an overview of the framework. The experimental design and the obtained results are described in Sections (IV and V). Finally, Section (VI) concludes the paper.

II. Related Work

A. Grounding

Grounding indicates the assignment of meaning to an abstract symbol, e.g. a word, through perceptual information [18]. Previous studies that investigated the use of cross-situational learning for grounding of objects [13, 40] as well as spatial concepts [2, 10, 41] ensured that one word appears several times together with the same perceptual feature vector so that a corresponding mapping can be created [14]. However, natural language is 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. The latter does not need to be actual synonyms, especially, considering that according to the "Principle of Contrast" no two words refer to the exact same meaning, i.e. there are no true synonyms [7]. Consequently, words are only synonyms as references to an object or action in a particular set of situations. Examples are words that refer to the purpose or content of an object, instead of the object itself, such as: *tea* or *coffee* instead of *cup*. Roesler et al. [33] proposed a cross-situational learning model for grounding of synonyms, however, they only considered *known* synonyms, i.e. they ensured that all synonyms were encountered during training.
In this study, we take a step towards grounding *unkown* synonyms, i.e. synonyms the model has never encountered before, through an unsupervised approach so as to infer the meaning of objects and actions.

B. Word Similarity

Determining similarity between words (lexical or semantic) is important for a variety of tasks such as word-sense disambiguation [28], and automatic evaluation of text summarisation [23] and machine translation [32]. Lexically similar words have similar character sequences, while semantically similar words have similar or opposite meanings [17]. For object and action grounding, only the semantic similarity is relevant. Semantic similarity can be either corpus- or knowledge-based. Our goal is to obtain semantic knowledge automatically in an unsupervised manner from plain text. Therefore, knowledge-based semantic similarities are not usable because most sophisticated knowledge representations are manually created [39]. Automatically assembled knowledge representations, on the other hand, provide only very simple relations like ordinate or component relations and allow only structured sources as input [39]. Due to the latter, a corpus-based method called Word2Vec is used in this study in order to allow the use of plain and unstructured text. Word2Vec uses a large corpus of plain text as input and outputs a vector space, where each distinct word is represented by a vector [25, 27]. The distance between two vectors corresponds to the syntactic-semantic similarity between two corresponding words. Thereby, Word2Vec defines syntactic-semantic relations between words implicitly by their locations in the vector space.

III. System Overview

The used grounding system consists of five parts: (1) Neural Network Language Model (Word2Vec), which creates a vector space in which the distance between two vectors represents their syntactic-semantic similarity, (2) Part-of-Speech (POS) tagging system, which gramatically tags words in an unsupervised manner (i.e., *it does not use any pre-tagged corpus or tagging dictionary to assign numerical tags to words*), (3) 3D object segmentation system, which determines the geometric characteristics of objects by segmenting them into point clouds, (4) Action recording system, which creates action feature vectors by recording the state of several joints while the robot is executing actions, and (5) Multimodal probabilistic learning model, which grounds object and action names through visual perception and proprioception. The inputs and outputs of the individual parts are highlighted below, and described in detail in the following subsections.

1) **Word2Vec**:
   - **Input**: Sentences, which represent the instructions given by the human tutor to the robot.[1]
   - **Output**: 9-dimensional real-valued vectors. The distance between two vectors represents the syntactic-semantic similarity between their corresponding words.
2) **POS tagging**
   - **Input**: Sentences, which represent the instructions given by the human tutor to the robot.
   - **Output**: Numerical tags. If two words have the same tag, they belong to the same syntactic category.
3) **3D object segmentation**:
   - **Input**: Point cloud data.
   - **Output**: Geometric characteristics of objects.
4) **Action recording**:
   - **Input**: Changes of joint states during the execution of actions by the robot.
   - **Output**: Action feature vectors representing the executed actions.
5) **Multimodal probabilistic learning model**:
   - **Input**: The outputs of 1, 2, 3 and 4.
   - **Output**: For a given test sentence and the corresponding feature vectors, the model determines the modality of each word.

A. Syntactic-Semantic Representation of Words

A Neural Network Language Model (NNLM) can be used to represent words as high-dimensional real-valued vectors. The literature reveals several different NNLM architectures [4,

[1]The instructions of the human tutor are given as input to Word2Vec during the experiments, i.e. after Word2Vec has been trained with 100MB of Wikipedia articles (Section III-A).

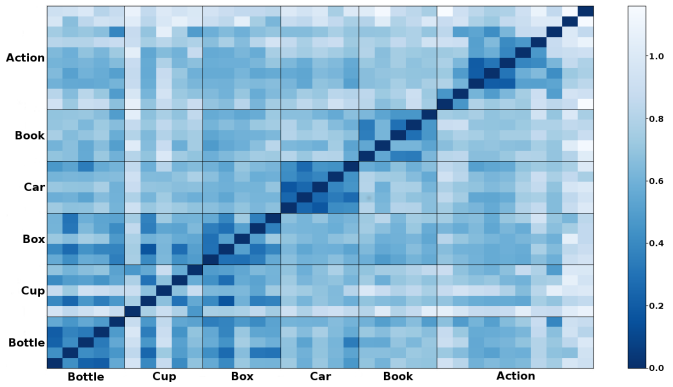


Fig. 2: Euclidean distances between the nine principal component vectors of all object and action names. The 5 vectors of each object represent the 5 corresponding names, while the 10 action vectors represent the 10 action names used in this study. The individual names can be obtained from Tables (I and II). At the top and bottom left corner are the names *shift* and *coca_cola*, respectively.

24, 37]. One of the main advantages of these models is the level of generalization, which is not possible to attain with simple n-gram models [27]. One recently developed NNLM is Word2Vec, which uses a 2-layer neural network to create word embeddings, i.e. a vector space, for a given text corpus. Words that are syntactically and semantically similar are located close together [25, 26]. Word2Vec was trained using 100MB of Wikipedia articles[2]. Several names used in this study are bigrams, i.e. they consist of two words, which would lead to two separate word vectors. Therefore, the original bigrams have been converted into unigrams by inserting an underscore between the two words as shown in Tables (I and II).
The vector space generated by Word2Vec has been transformed to an Euclidean space using MDS [8]. Afterwards, PCA has been applied to reduce the high vector dimensionality (100 dimensions) so as to efficiently ground vectors in perception. The resulting Euclidean distances between all names are shown in Figure (2). The figure reveals that the vectors referring to *Car* and *Book* names are well clustered into two separate groups, while the vectors referring to *Bottle*, *Cup* and *Box* names are clustered together. The latter is due to the fact that they refer to container-type objects for food or drinks. Vectors referring to action names are grouped into one cluster so that the individual action name pairs (Table II), such as (*lift_up*, *raise*) and (*grab*, *take*) are not separated into five independent groups, which shows that the syntactic-semantic information provided by Word2Vec is not highly descriptive. These findings are consistent with the results indicated in Table (III), which shows that the mean distances between word vectors of the same modality, e.g. *object-object*, is *slightly* smaller than the mean distances between word vectors of the **Object** and **Action** modalities.

[2]The corpus can be downloaded at http://mattmahoney.net/dc/text8.zip.

TABLE I: Overview of the objects with their corresponding synonyms.

Object	**Synonyms**				
Bottle	coca_cola	soda	pepsi	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	the_godfather	narnia	lord_of_the_rings	the_hobbit

TABLE II: Overview of the used actions.

Synonym 1	**Synonym 2**	**Description**
lift_up	raise	The object will be lifted up.
grab	take	The object will be grabbed, but not displaced.
push	poke	The object will be pushed with the closed gripper i.e. it will not be grabbed.
pull	drag	The object will be grabbed and moved towards the robot.
move	shift	The object will be grabbed and moved.

B. *Syntactic Representation of Words*

Part-of-Speech (POS) tagging marks words in sentences with grammatical attributes (e.g., noun, verb, adjective, etc.). A variety of supervised, semi-supervised, and unsupervised POS tagging approaches exist in the literature [5, 6, 42]. In this study, grammatical tags are induced for word sequences in an unsupervised manner through a first-order Bayesian Hidden Markov Model (HMM), i.e., *without using any pre-tagged training corpus*[3]. A grammatical tag $\mathrm{T} = (t_1, \ldots, t_n)$ is assigned to each word in the sequence $\mathrm{W} = (\omega_1, \ldots, \omega_n)$ by the POS tagging model, which uses words as observations and tags as hidden states (Figure 3) [15].

The probability distribution of tag states for the word sequence w is defined as follows:

$$\mathbb{P}(t_1, \ldots, t_n) = \prod_{i=1}^{n} \mathbb{P}(t_i \,|\, t_{i-1}) \tag{1}$$

where the transition probability to the tag t_i is conditioned on the tag t_{i-1}. This could encode the intuitive grammar that parts of speech might follow, like having a noun after a determiner. Emission distributions of numerical tags over words are defined through the probability $\mathbb{P}(\omega_i \,|\, t_i)$ of the word ω_i being conditioned on the tag t_i. For each tag state the generative transition and emission parameters of the proposed HMM model (ϕ, θ) are characterized through multinomial distributions with Dirichlet priors $(\alpha_\phi, \alpha_\theta)$ (where K denotes the number of tag states):

$$\begin{aligned} t_i \,\big|\, t_{i-1} = t \sim \mathit{Mult}\,(\phi_t) \quad , \quad \phi_t \,\big|\, \alpha_\phi \sim \mathit{Dir}\,(\alpha_\phi) \\ \omega_i \,\big|\, t_i = t \sim \mathit{Mult}\,(\theta_t) \quad , \quad \theta_t \,\big|\, \alpha_\theta \sim \mathit{Dir}\,(\alpha_\theta) \end{aligned} \tag{2}$$

For an unannotated training corpus containing a set of m sentences $\mathrm{W} = \{w_1, \ldots, w_m\}$, the POS tagging model tries to induce the most likely numerical tag set $\mathrm{T} = \{\tau_1, \ldots, \tau_m\}$

[3]For example, the POS tagging system could assign these numerical tags to words of the sentence: (Push,**7**) (the,**5**) (Coffee,**9**).

TABLE III: Mean intra- and inter-modality distances between word vectors. The intra-modality distance for the "Others" modality is zero because *only one word* (the article *the*) belongs to it, which is not sufficient to create an independent cluster.

	Object	Action	Others
Object	0.61	0.75	0.98
Action	0.75	0.55	1.01
Others	0.98	1.01	0.00

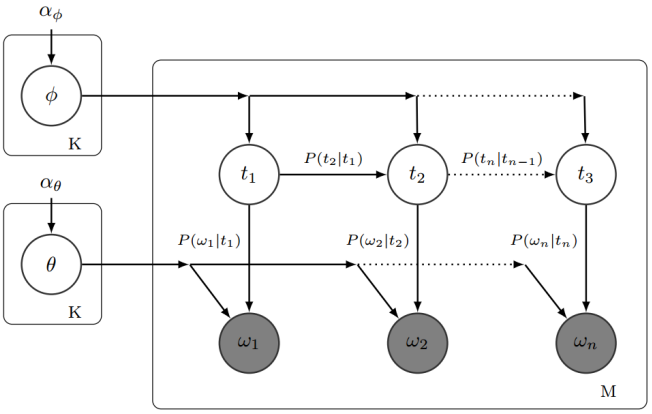


Fig. 3: Graphical representation of the HMM-based Part-of-Speech tagging model.

for each sentence in the corpus that maximizes the following expression:

$$\mathbb{P}(T,W) = \prod_{(\tau,w)\in(T,W)} \Big(\mathbb{P}(T,w\,|\,\phi,\theta)\Big) = \prod_{(\tau,w)\in(T,W)} \Big(\prod_{i=1}^{n} \mathbb{P}(t_i\,|\,t_{i-1},\phi_t)\mathbb{P}(\omega_i\,|\,t_i,\theta_t)\Big) \tag{3}$$

Inferring the latent tag variables uses the Gibbs sampling algorithm [16, 29], which produces a set of samples from the posterior distribution $\mathbb{P}(T|W)$, i.e., it loops over the possible tag assignments to words that could maximize (3) expressed as follows, where *-i* denotes all samples except the *i*-th sample:

$$\mathbb{P}(T_i, T^{(i)} \,\big|\, T_{-i}, W, T^{(-i)}, w, \alpha_\phi, \alpha_\theta) \tag{4}$$

C. 3D Object Features

The object feature vectors are obtained using 3D point cloud segmentation [30]. Different segmentation approaches have been investigated in the related literature. Edge based methods segment point clouds into regions by detecting their boundaries, which are characterized by points with a fast intensity change [35]. These methods are fast, but also highly sensitive to noise. Region based methods determine regions by combining neighbouring points that have similar properties [22]. They are less susceptible to noise, but are not good at determining exact region borders. Attributes based methods use predefined attributes, such as point density and vertical

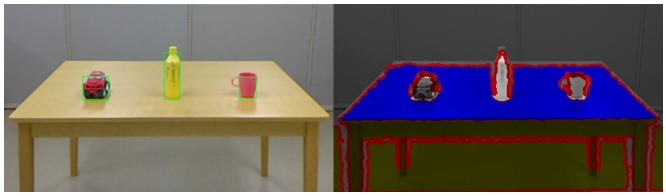


Fig. 4: Examples of the used objects and the corresponding 3D point cloud information: (A) car, (B) bottle, and (C) cup.

distribution, to cluster point clouds [11]. These methods can be very accurate and flexible, but they are often slow and the overall performance depends heavily on the quality of attributes. Graph based methods treat point clouds as a graph, where each point represents a vertex connected via edges to neighbouring points [38]. These methods can handle data with noise or uneven density, but they can not often be run in real time. Model based approaches use primitive geometric shapes in order to create clusters of points with similar mathematical representations [36]. They are fast and can handle outliers, however, they are inaccurate when dealing with point clouds from different sources.

In this study, a model based segmentation approach is used due to its speed, reliability, and the fact that no much prior knowledge about the environment is required, such as object models and the number of regions to process [9]. The applied model detects the major plane in the environment[4] via the RANSAC algorithm [12], and keeps track of it in consecutive frames. Planes that are orthogonal to the major plane and touch at least one border of the image are defined as wall planes, while points that are neither part of the major nor the wall planes are voxelized and clustered into blobs. Blobs of reasonable size, i.e. neither extremely small nor large, are treated as objects[5]. Each point cloud of a segmented object is characterized through a Viewpoint Feature Histogram (VFH) [34] descriptor, which represents the geometry of the object taking into account the viewpoint and ignoring scale variance. Figure (4) shows an example of the obtained 3D point cloud information.

D. Action Features

Action feature vectors were formulated to represent the dynamic characteristics of actions during execution through teleoperation, which could afford variations in the obtained action feature vectors[6]. Overall, five different characteristics - each representing a possible subaction - are recorded using the sensors of the robot [43]. The employed characteristics are:

1) The distance from the actual to the lowest torso position in meters.
2) The angle of the arm in radians.
3) The angle of the wrist in radians.

[4]The major plane in the conducted experiment is a tabletop.

[5]The threshold was manually set after selecting the objects for the experiment and should be suitable for all objects of similar size.

[6]While our focus in this study is on grounding recorded actions, we will investigate the use of the learned model to generate actions in the future [19].

TABLE IV: Definitions of learning parameters in the graphical model.

Parameter	Definition
λ	Hyperparameter of the distribution π_w
α_g	Hyperparameter of the distribution π_g
α_a	Hyperparameter of the distribution π_a
m_i	Modality index of each word. (modality index $\in$ {Object, Action, Others})
Z_a	Index of action feature vector distributions
Z_g	Index of object geometry distributions
w_i	Word vectors, POS tags, or Word indices
g	Observed state representing geometric characteristics of object using VFH descriptor
a	Observed state representing characteristics of action
γ	Hyperparameter of the distribution $\theta_{m,Z}$
β_a	Hyperparameter of the distribution ϕ_a
β_g	Hyperparameter of the distribution ϕ_g
$\theta_{m,Z}$	Word distribution over modalities

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 difference in the angles of the arm and wrist in radians, respectively. The differences are calculated by subtracting the value at the beginning of the subaction from the value 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 subactions, which have been manually defined. Consequently, if an action consists of less than six subactions, rows with zeros will be added at the end. The length, i.e. duration, of a subaction depends on the teleoperator and is therefore not fixed.

E. Probabilistic Learning Model

The process of grounding object and action names through perception employs a Bayesian learning model as outlined in Figure (5). A probabilistic graphical model is a directed acyclic graph representing a set of probability distributions that can handle uncertainty represented by noisy perceptual data obtained from the environment [20]. Four different word representations are used for the Bayesian learning model: (1) **Word indices, i.e. each word is represented by a different number**[7], (2) POS tags, i.e. words are represented by the grammatical categories they belong to, (3) Word vectors, i.e. words are represented by vectors in a syntactic-semantic vector space, and (4) Syntactic-semantic vectors and POS tags. When words are represented by Indices or POS tags **categorical** and **Dirichlet** distributions are used for w_i and $\theta_{m,Z}$, respectively. If words are represented by syntactic-semantic vectors **Gaussian** and **Gaussian Inverse-Wishart** distributions are used instead.

[7]For example, the following indices could be assigned to words of the sentences: (Push, **1**) (the, **2**) (Coffee, **3**) and (Pull, **4**) (the, **2**) (Milk, **5**). Even though *Push* and *Pull* belong to the same category, i.e. verb, they have different indices. Unlike the case of POS tags where the assigned tags would be: (Push, **1**) (the, **2**) (Coffee, **3**) and (Pull, **1**) (the, **2**) (Milk, **3**).

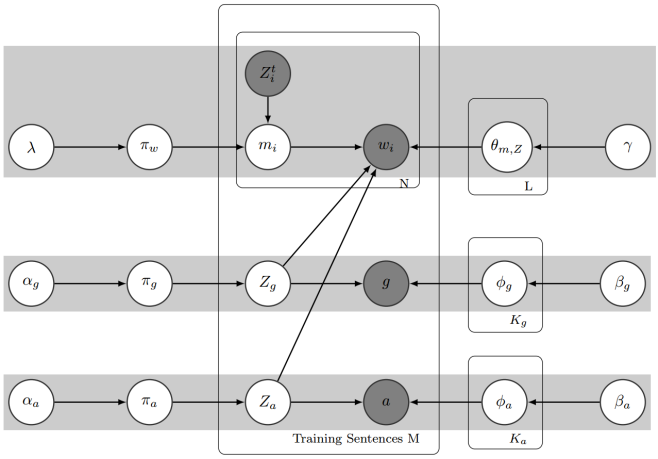


Fig. 5: Graphical representation of the probabilistic model. Indices **i**, **g** and **a** denote the order of words, object geometric features and action features, respectively.

In the probabilistic learning model, the observed state w_i represents word indices, syntactic-semantic vectors, or POS tags (Sections III-A and III-B). The observed state g represents the geometric characteristics of objects expressed through the VFH descriptor (Section III-C). Actions are represented by the observed state a (Section III-D). The observed state Z_i^t only exists in the model that combines word vectors and POS tags, in that case, w_i represents syntactic-semantic vectors and Z_i^t represents the corresponding POS tags. Table (IV) provides a summary of the definitions of the learning model parameters. The corresponding probability distributions, i.e., w_i, $\theta_{m,Z_{L_1}}$, $\phi_{a_{K_1}}$, $\phi_{g_{K_2}}$, π_w, π_g, π_a, m_i, Z_g, Z_a, g, and a, which characterize the different modalities in the graphical model, are defined in (5), where N denotes a multivariate Gaussian distribution, *GIW* denotes a Gaussian Inverse-Wishart distribution, *Dir* denotes a Dirichlet distribution, and *Cat* denotes a categorical distribution.

$$\begin{cases} w_i & \sim & Cat(\theta_{m_i,Z_{m_i}}) \\ & & N(\theta_{m_i,Z_{m_i}}) \\ \theta_{m,Z_{L_1}} & \sim & Dir(\gamma) \\ & & GIW(\gamma) \quad , \quad L_1 = (1,...,L) \\ \phi_{a_{K_1}} & \sim & GIW(\beta_a) \quad , \quad K_1 = (1,...,K_a) \\ \phi_{g_{K_2}} & \sim & GIW(\beta_g) \quad , \quad K_2 = (1,...,K_g) \\ \pi_w & \sim & Dir(\lambda) \\ \pi_g & \sim & Dir(\alpha_g) \\ \pi_a & \sim & Dir(\alpha_a) \\ m_i & \sim & Cat(\pi_w) \\ Z_g & \sim & Cat(\pi_g) \\ Z_a & \sim & Cat(\pi_a) \\ g & \sim & N(\phi_{Z_g}) \\ a & \sim & N(\phi_{Z_a}) \end{cases} \tag{5}$$

The latent variables of the Bayesian learning model are inferred using the Gibbs sampling algorithm [16], which repeatedly samples from and updates posterior distributions.

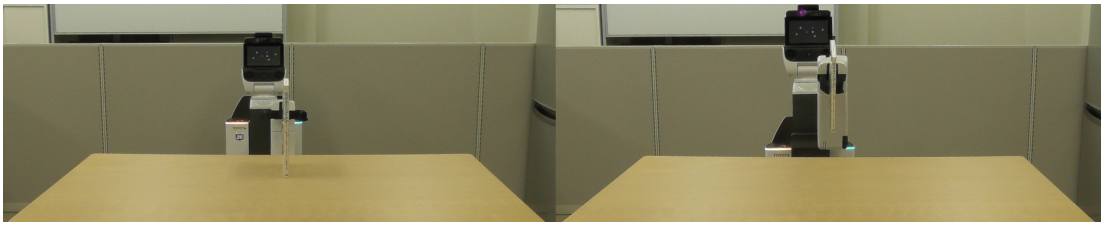


Fig. 6: Illustration of action *lift up* executed by the robot in a tabletop scene.

IV. Experimental Setup

A human tutor and HSR robot[8] are interacting in front of a tabletop. The robot does not have any preexisting knowledge about the world, and its syntactic-semantic knowledge is limited to the word vector space and the HMM-based POS tagging model, which were created prior to training. One of the five different objects {BOTTLE, CUP, BOX, CAR, and BOOK} is placed on the table (Figures 4 and 6). Each of the objects can be referred to by five different names as shown in Table (I). During the cross-situational learning phase [13], the robot performs five different actions on each object (Figure 6), where each action can be described by two different names as illustrated in Table (II).

A total of 75 different sentences are given to the robot by the human tutor in order to allow it to ground object and action names using the recorded perceptual data. Each sentence consists of either two or three words and has one of the following two structures: "*action* the *object*" or "*action object*", respectively[9], where *action* and *object* are substituted by the corresponding names (Tables I and II). The experimental procedure consists of three phases as described below:

1) Collection of semantic and perceptual information for the different situations.
   a) An object is placed on the table and the robot determines its geometric characteristics so as to calculate its feature vector (Section III-C).
   b) A sentence is given by the human tutor to the robot, and the corresponding vector and POS tag of each word is obtained (Sections III-A and III-B).
   c) The human tutor teleoperates the robot to execute the given action while several kinematic characteristics are recorded and converted into an action feature vector (Section III-D).
2) The probabilistic model is used to ground words using the geometric characteristics of objects and the action feature vectors (Section III-E).
3) For the test phase, a total of 50 sentences are used to evaluate the learning framework.

In this study, none of the object and action names used during the test phase are part of the training sentences to allow investigating the capability of the Bayesian learning model to ground unknown synonyms.

V. Results and Discussion

In several previous studies, probabilistic models have been used for language grounding [1, 10, 41]. However, *to the best of our knowledge*, none of them included *unknown* synonyms and they differed in their approaches, experimental setups, or corpora from the current study, *which makes the comparison of results between our study and these studies, among many others in the literature, difficult to attain*. 20 fold cross-validation has been used, i.e. 20 different training and test sets have been created. 75 sentences have been used for training, while the remaining 50 sentences have been used for the test phase.

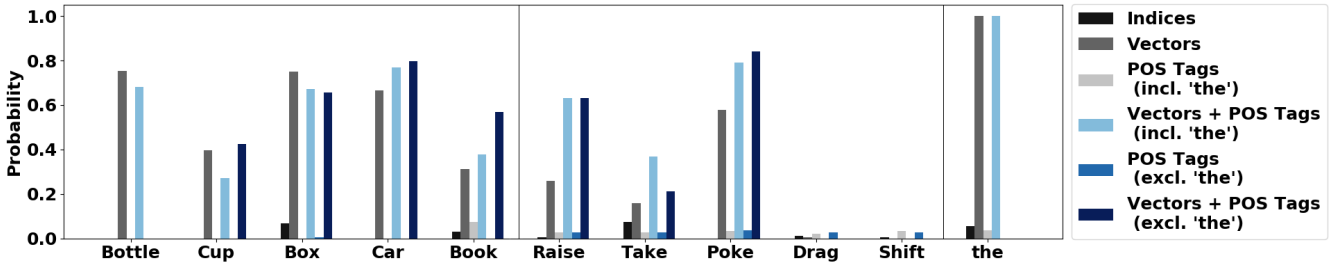
Four different word representations have been investigated (Section III-E). The obtained F1-scores show that combining syntactic-semantic vectors and POS tags achieves the best overall grounding performance with and without the article *the* (Figure 8) [10]. For the **Word Vector + POS Tags** and **Word Vector** representations the model did not learn the **Others** modality (the article *the*), which might be due to the inter- and intra-modality distances in the employed vector space (Table III). While the intra-modality distances for **Object** and **Action**, e.g. *object-object*, and the corresponding inter-modality distance, i.e. *object-action*, are very similar, the inter-modality distances between the **Others** and the **Object** as well as **Action** modalities is much higher. The larger distance seems to prevent the model from learning the **Others** modality correctly, in addition to that the **Others** modality only contains one word (the article *the*), which is, theoretically, not sufficient to create an independent cluster for the learning model. This conclusion is supported by a test where we clustered word vectors and gave cluster labels as input to the Bayesian learning model[11]. Using cluster labels instead of word vectors

[8]The Human Support Robot from Toyota is used for the experiment. It has a cylindrical shaped body, which can move omnidirectional, and is equipped with one arm and a gripper to grasp objects. The robot has 11 degrees of freedom and is equipped with stereo and wide-angle cameras, a microphone, a display screen, and a variety of different sensors. [Official Toyota HSR Website]

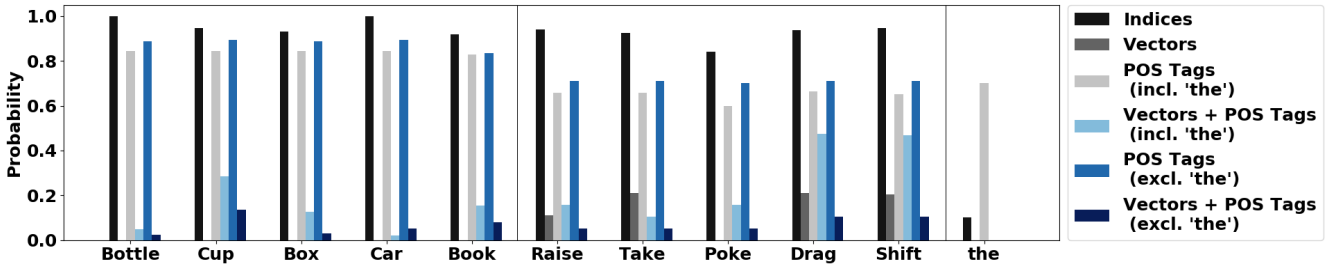
[9]The latter is only used for sentences with the BOOK object. For example: "LIFT_UP HARRY_POTTER" represents the structure "*action object*", while "LIFT_UP the LEMONADE" represents the structure "*action* the *object*".

[10]These results are independent of the exact values of the model hyperparameters because they do not have much influence on the grounding performance, which is why detailed results of conducted parameter testing are not included in the paper.

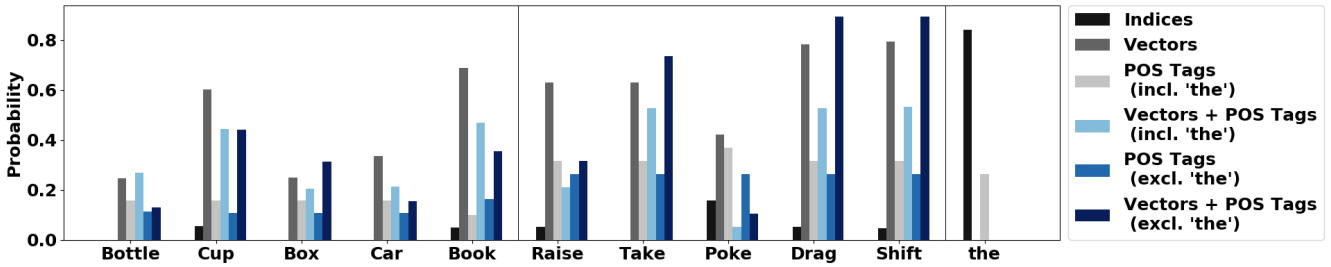
[11]When cluster labels are used, words that belong to the same cluster are represented by the same label.



(a) Probabilities of word categories for the **Object** modality for all employed word representations.



(b) Probabilities of word categories for the **Action** modality for all employed word representations.



(c) Probabilities of word categories for the **Others** modality for all employed word representations.

Fig. 7: Probability distributions of word categories over the different modalities. The names for the different object and action categories are shown in Tables (I and II).

enabled the model to learn the **Others** modality with a F1-score of 0.2504. The use of cluster labels ensures that the distance between the different modalities is always the same, which is not the case for the **Others** modality when using word vectors. The article *the* was mostly classified as an object because the distance between the **Others** and **Object** modality is slightly smaller than the distance between the **Others** and **Action** modality (Table III).

When the **POS Tags** representation was used, the model did not learn the **Object** modality and assigned most words to the **Action** modality (Figure 7). One might argue that this performance is due to the relatively short sentences and the fact that objects occurred in two different positions in the two employed sentence structures, i.e. "*action* the *object*" or "*action object*", which did not allow the first-order Bayesian HMM (Section III-B) to learn the syntactic category of the **Object** modality with respect to the previous parts of speech. However, this is not the case because removing the article *the* from all sentences to ensure that all sentences have the same structure, i.e. "*action object*", did not allow the model to learn the **Object** modality. Thereupon, a logical explanation could be that the short sentences did not allow the model to create enough samples for learning [16]. This hypothesis is concordant with the findings of Aly et al. [3], where longer sentences with more than one object allowed a similar model to appropriately learn the **Object** modality[12]. However, this requires further investigations through a different experimental setup and a new study.

The **Word Indices** representation was used as a baseline because it uses neither syntactic nor semantic information. Since none of the object and action names of the test set occurred during training, their indices were not grounded. As a result, the model assigned most object and action names to the **Action** modality, which lead to a very low F1-score for **Object** and relatively high F1-score for **Action** caused by a precision and recall of around 0.5 and 1.0. For the **Others** modality the F1-score was high because the index of the article *the* appeared during training. The performance of the baseline model illustrates that unknown synonyms cannot be grounded without some kind of syntactic or semantic information.

[12]While the sentences used by Aly et al. [3] are longer, the employed short sentences in our current study are more **intuitiv** and near to the **daily language used by human users** to interact with robots, which constituted our motivation for the experimental setup of this study.

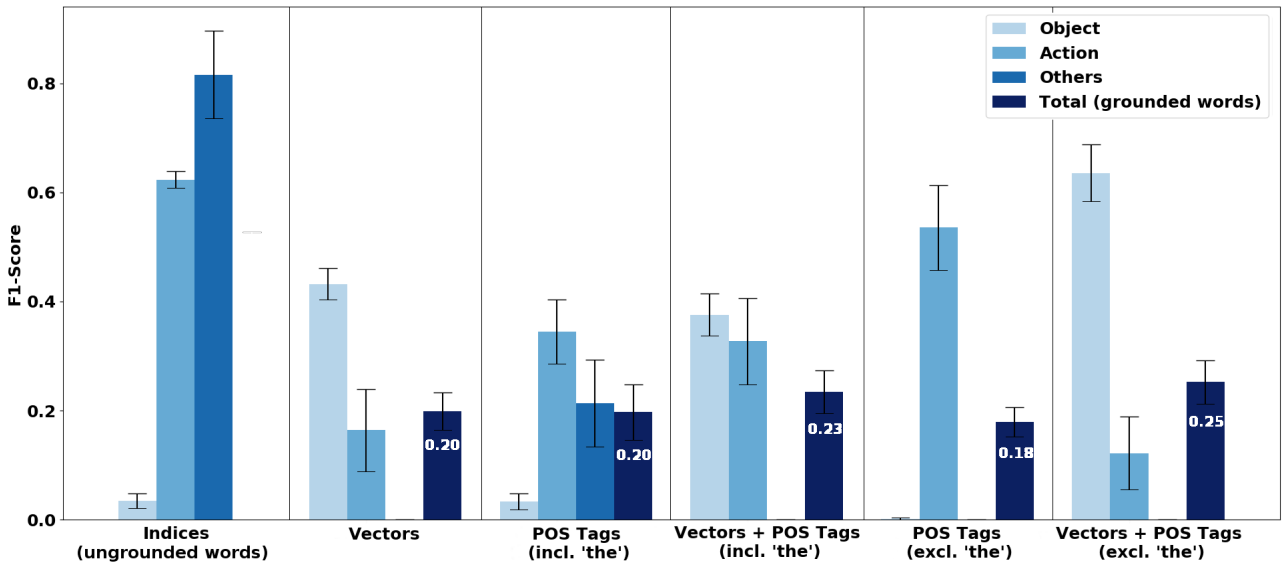


Fig. 8: Mean F1-scores for the different modalities and models. For the total F1-score bars, the corresponding numerical values are shown in the upper parts. The mean F1-scores are calculated over all the 20 folds of cross-validation, while the total F1-scores are the mean values of the F1-scores of all three modalities. The **Word Indices** model uses neither syntactic nor semantic information; therefore, it does not ground any word.

The results show that the syntactic-semantic information provided by word vectors and syntactic information represented by POS tags allow for grounding of unknown synonyms and that combining both achieves the best performance. Although achieving better grounding by adding additional information seems intuitive, it differs from the results of a previous study by Roesler et al. [33], where they showed that adding syntactic and semantic information negatively affects the grounding of *known* synonyms, while the best grounding was achieved using word indices. Therefore, further investigations seem to be necessary to understand why using more information has opposite effects for *known* and *unknown* synonyms and to develop a model that achieves the best grounding in both cases. On the one hand, representing semantic and syntactic information in a different form might avoid the negative effect on grounding of known synonyms. On the other hand, it might be possible to combine the models so that word indices are used for grounding, when synonyms are known, while word vectors and POS tags are used for unknown synonyms.

VI. Conclusions and Future Work

We investigated a multimodal framework for grounding *unknown* synonymous object and action names through the robot visual perception and proprioception during its interaction with a human tutor. Our Bayesian learning model was set up to learn the meaning of object and action names using geometric characteristics of objects obtained from point cloud information and kinematic features of the robot joints recorded during action execution.

The proposed model allowed the grounding of *unknown* synonyms based on syntactic and semantic information provided by POS tags and word vectors. The former were obtained through a HMM-based POS tagger, while the latter were obtained via Word2Vec, a neural network language model. Although the used syntactic and semantic information made grounding possible in general, there is still opportunity to further enhance the grounding. For example, by improving the employed word embeddings, which can be achieved by employing a larger or more domain specific corpus to create the vector space.

In future work, the proposed learning model will be extended to work *online*, i.e. it will be able to update its learning parameters in case of new objects and actions. Furthermore, we will extend the model so as to include other modalities such as color, in addition to handling *known* synonyms. Finally, we will investigate the use of the model in generating actions and learning more complex sentence structures and a larger number of words. This constitutes a future research direction of the current study.

References

[1] A. Aly and T. Taniguchi. 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, May 2018.

[2] A. Aly, A. Taniguchi, and T. Taniguchi. 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.

[3] A. Aly, T. Taniguchi, and D. Mochihashi. A probabilistic approach to unsupervised induction of combinatory categorial grammar in situated human-robot interaction. In *IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, Beijing, China, November 2018.

[4] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research (JMLR)*, 3(6):11371155, 2003.

[5] E. Brill. A simple rule-based part of speech tagger. In *Proceedings of the 3rd Conference on Applied Natural Language Processing (ANLC)*, Trento, Italy, 1992.

[6] E. Brill and M. Pop. Unsupervised learning of disambiguation rules for part-of-speech tagging. In S. Armstrong, K. Church, P. Isabelle, S. Manzi, E. Tzoukermann, and D. Yarowsky, editors, *Natural Language Processing Using Very Large Corpora*, volume 11 of *Text, Speech, and Language Technology*, page 2742. Springer, 1999.

[7] E. V. Clark. The principle of contrast: A constraint on language acquisition. In *Mechanisms of Language Acquisition*, pages 1–33. Lawrence Erlbaum Associates, 1987.

[8] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*. Chapman and Hall, 2001.

[9] C. Craye, D. Filliat, and J.-F. Goudou. Environment exploration for object-based visual saliency learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, May 2016.

[10] C. R. Dawson, J. Wright, A. Rebguns, M. V. Escárcega, D. Fried, and P. R. Cohen. 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.

[11] S. Filin and N. Pfeifer. Segmentation of airborne laser scanning data using a slope adaptive neighborhood. *ISPRS Journal of Photogrammetry & Remote Sensing (P&RS)*, 60:71–80, 2006.

[12] M. A. Fischler and R. C. Bolles. 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, June 1981.

[13] J. F. Fontanari, V. Tikhanoff, A. Cangelosi, R. Ilin, and L. I. Perlovsky. Cross-situational learning of object-word mapping using neural modeling fields. *Neural Networks*, 22(56):579–585, JulyAugust 2009.

[14] J. F. Fontanari, V. Tikhanoff, A. Cangelosi, and L. I. Perlovsky. 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.

[15] J. Gao and M. Johnson. A comparison of Bayesian estimators for unsupervised hidden markov model pos taggers. In *Proceedings of the 13th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page 344352, Honolulu HI, USA, 2008.

[16] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 6(6):721–741, November 1984.

[17] W. H. Gomaa and A. A. Fahmy. A survey of text similarity approaches. *International Journal of Computer Applications (IJCA)*, 68(13):13–18, April 2013.

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

[19] T. Inamura, I. Toshima, H. Tanie, and Y. Nakamura. Embodied symbol emergence based on mimesis theory. *The International Journal of Robotics Research (IJRR)*, 23(4-5):363–377, April 2004.

[20] M. I. Jordan. Graphical models. *Statistical Science*, 19 (1):140155, 2004.

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

[22] K. Koster and M. Spann. Mir: An approach to robust clustering-application to range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22(5):430–444, May 2000.

[23] C.-Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the Human Technology Conference (HLT-NAACL)*, Edmonton, Canada, May 2003.

[24] T. Mikolov, M. Karafiat, J. Cernocky, and S. Khudanpur. Recurrent neural network based language model. In *Proceedings of Interspeech*, Makuhari, Chiba, Japan, September 2010.

[25] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *ArXiv e-prints*, January 2013. eprint: 1301.3781.

[26] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *ArXiv e-prints*, October 2013. eprint: 1310.4546.

[27] T. Mikolov, W. t. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-2013)*. Association for Computational Linguistics, May 2013.

[28] R. Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2), February 2009.

[29] G. Neubig. Simple, correct parallelization for blocked gibbs sampling. Technical report, Nara Institute of Science and Technology, November 2014.

[30] A. Nguyen and B. Le. 3D point cloud segmentation: A survey. In *6th IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, Manila, Philippines, November 2013. IEEE.

[31] International Federation of Robotics. World robotics

2017 - service robots, October 2017.

[32] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318, Philadelphia, July 2002.

[33] O. Roesler, A. Aly, T. Taniguchi, and Y. Hayashi. 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.

[34] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu. 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, October 2010.

[35] A. D. Sappa and M. Devy. Fast range image segmentation by an edge detection strategy. In *Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling (3DIM)*, Quebec City, Quebec, Canada, August 2002.

[36] R. Schnabel, R. Wahl, and R. Klein. Efficient ransac for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226, June 2007.

[37] H. Schwenk. Continuous space language models. *Computer Speech and Language*, 21(3):492518, 2007.

[38] J. Strom, A. Richardson, and E. Olson. Graph-based segmentation for colored 3D laser point clouds. In *International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 2010.

[39] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A core of semantic knowledge unifying wordnet and wikipedia. In *Proceedings of the 16th international conference on World Wide Web (WWW 2007)*, pages 697–706. ACM, 2007.

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

[41] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. Teller, and N. Roy. Approaching the symbol grounding problem with probabilistic graphical models. *AI Magazine*, 32(4):6476, 2011.

[42] K. Toutanova and M. Johnson. A Bayesian LDA-based model for semi-supervised part-of-speech tagging. In *Proceedings of the 20th International Conference on Neural Information Processing Systems (NIPS)*, page 15211528, Vancouver, Canada, 2007.

[43] *HSR Manual*. Toyota Motor Corporation, 2017.4.17 edition, April 2017.