DCG-UPUP-Away: Automatic Symbol Learning through Grounding to Unknowns

Mycal Tucker, Derya Aksaray, Rohan Paul, Gregory J. Stein, and Nicholas Roy

***Abstract*— This work addresses the symbol grounding problem, that is, understanding the "meaning" of natural language within a robot's workspace. The state of the art models used in the grounding problem typically assume a fixed set of phrases or objects that are defined a priori to mission. However, the real world is full of unexpected objects that are nearly impossible to anticipate and therefore train for. This paper proposes a model called the "Distributed Correspondence Graph - Unknown Phrase, Unknown Percept - Away" that explicitly represents unknown phrases and objects as unknown symbols and enables to reason about objects outside the field of view. Moreover, the model is capable of acquiring new symbols in an online fashion. The effectiveness of the model is evaluated via simulations and real experiments in terms of grounding and learning new phrases and objects.**

I. Introduction

Recently, there has been a great interest in human-robot teaming in civilian (e.g., at factories, hotels, hospitals, homes) and military (e.g., reconnaissance) applications. Communication plays an important role in effective teaming between humans and robots, and communication via natural language provides a rich, intuitive, and flexible medium. Accordingly, the grounding problem in the literature addresses the question of how a robot can understand the meaning of a natural language command in the context of its world model (e.g., [1], [2], [3]).

The existing methods for solving the grounding problem make two primary assumptions. First, they assume a fixed set of phrases that can constitute the commands and a fixed set of objects that exist in the world model. Such methods typically fail to reason about unknown phrases or objects that have never been encountered in the training process. Second, the location of the object being grounded is often assumed to be known. In other words, the phrases are assumed to refer to objects that are currently perceived or localized within a known map. As a result, a robot using these methods tends to pick the most likely perceived grounding rather than exploring its surroundings even for unknown words.

Note that such assumptions may not be valid in real-world applications. For example, humans tend to use context-specific lexicons in their daily life, or they often refer to objects whose locations may be unknown. To deal with such cases, training a robot to know the meaning of every possible word is infeasible and inefficient. Also, attempting to reason over the space of all possible maps is similarly computationally infeasible.

All authors are with the Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology, Cambridge, MA 02139, USA {mycal,daksaray,rohanp,gjstein,nickroy}@csail.mit.edu

This work was partially supported by the Robotics Consortium of the U.S Army Research Laboratory under the Collaborative Technology Alliance Program.

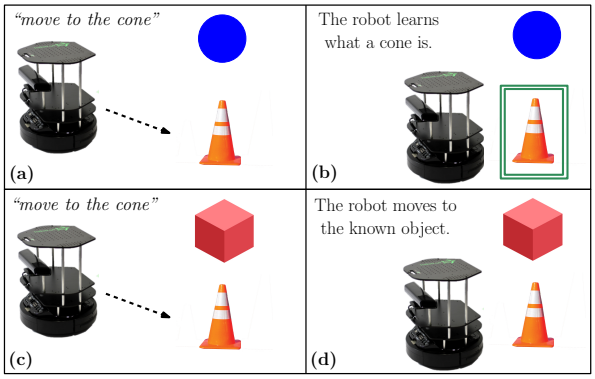


Fig. 1: An illustration of learning and grounding an unknown object. The robot (a) knows what a sphere is, (b) learns what a cone is, (c) sees a cone again, (d) moves towards the cone that is a known object from now on.

This paper proposes a model called the Distributed Correspondence Graph - Unknown Phrase, Unknown Percept - Away (DCG-UPUP-Away), which removes these assumptions by 1) explicitly modeling unknown phrases and percepts, 2) grounding hypothetical objects outside the field of view, and 3) allowing the robot to explore its surroundings to discover the meaning of unknown words. These changes yield a model that can ground a large variety of phrases in complex environments and facilitate learning new words and objects in an online fashion. The model is validated by a simulation study using commands generated by Amazon Mechanical Turk users. Also, the performance of the model is evaluated via real experiments where a TurtleBot is initially trained to recognize a small set of phrases and objects. The results demonstrate that the model provides a high grounding accuracy with new symbol acquisitions.

II. Grounding Natural Language Instructions

Grounding natural language addresses the problem of understanding the meaning of a command within the current world configuration containing numerous objects. For example, grounding the command "move to the cube" is inferring 1) the cube object in the world configuration, 2) the feasible region near the cube, and 3) the act of going towards the feasible region. Accordingly, let $\boldsymbol{\lambda}$ be the natural language command, let Γ be the set of groundings that correspond to semantic notions such as objects, locations, regions, paths,

or actions the robot can take, and let Υ denote the physical workspace of the robot that aggregates metric and semantic information about the constituent objects. Then, the symbol grounding problem can be formulated as

$$\boldsymbol{\gamma}^* = \arg\max_{\boldsymbol{\gamma} \in \Gamma^{|\boldsymbol{\lambda}|}} p(\boldsymbol{\gamma}|\boldsymbol{\lambda}, \Upsilon), \tag{1}$$

where $\boldsymbol{\lambda}$ is a vector of phrases from the set Λ and $\Lambda = \{\text{English phrases}\}$ denotes what phrases natural language sentences may be composed of; $|\boldsymbol{\lambda}|$ is the length of the command; $\boldsymbol{\gamma} \in \Gamma^{|\boldsymbol{\lambda}|}$ is a vector of groundings with a length of $|\boldsymbol{\lambda}|$; and the optimal vector of groundings $\boldsymbol{\gamma}^*$ is the one with maximum likelihood, given a command $\boldsymbol{\lambda}$ and a world model Υ.

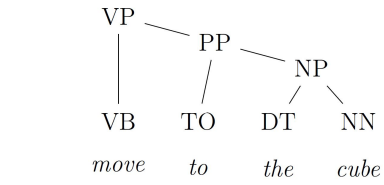
One efficient way of solving Eqn. (1) can be achieved via the Distributed Correspondence Graph (DCG) [2]. This model is structured according to the hierarchical structure of the language command. For example, the parse structure of the instruction "move to the cube" is illustrated in Fig. 2a. Accordingly, the goal becomes to find the correct associations between "move", "to", "the cube" and objects, regions in the world.

In the DCG model, the domains of Γ and Λ are defined a priori, and Υ represents the world model that contains the locations and identities of objects perceived by the robot. The set of phrases Λ is assumed to only contain words that have appeared in the training examples. Similarly, the set of grounding symbols Γ is assumed to contain 1) symbolic objects that the robot has been trained with, and 2) regions and motion constraints that are obtained by discretizing the perceived continuous space. Additionally, the DCG model introduces another set of variables Φ, and $\phi_{ij} \in \Phi$ is called a correspondence variable that refers to a binary relationship between the i^{th} phrase λ_i and the j^{th} grounding variable γ_{ij}. The main assumption of the DCG model is that the grounding variables are conditionally independent from each other given the phrases. Overall, the DCG solves an inference problem as a search over the unknown correspondence variables as follows:

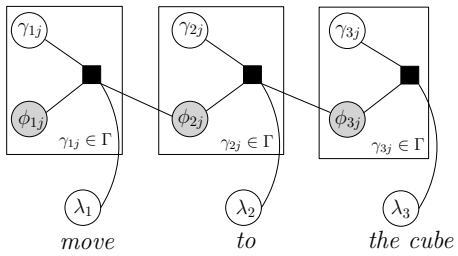
$$\boldsymbol{\phi}^* = \arg\max_{\phi_{ij} \in \boldsymbol{\phi}} \prod_{i}^{|\boldsymbol{\lambda}|} \prod_{j}^{|\Gamma^i|} p(\phi_{ij}|\gamma_{ij}, \lambda_i, \Phi_{c_i}, \Upsilon_{KP}), \tag{2}$$

where $\lambda_i \in \Lambda_{KN}$ and Λ_{KN} is the set of phrases with known (previously seen) words; Γ^i is the set of grounding variables of λ_i [1], ϕ_{ij} is the j^{th} correspondence variable of λ_i; γ_{ij} is the j^{th} grounding of λ_i; Υ_{KP} denotes the world model consisting of the set of known perceived symbolic objects and regions; and Φ_{c_i} is the set of child correspondence variables of λ_i. Note that the hierarchical structure of a command induces a parse tree (as in Fig. 2a), and the structure of the tree is used to instantiate the DCG model (as in Fig. 2b). In this context, Φ_{c_i} is defined as the set of correspondence variables for the immediate children phrases

[1]If λ_i is a noun phrase, the corresponding grounding set Γ^i contains the objects in the world (i.e., Γ^O). If λ_i is a verb phrase referring to the actions that the robot can take (e.g., "move", "pick"), then Γ^i contains the regions discretized with respect to the objects under consideration (i.e., Γ^{RO}).



(a) The parse tree for the command "move to the cube"



(b) The DCG graphical model for the parse tree in Fig. 2a

Fig. 2: An illustration of a parse tree and the corresponding DCG model where the graph topology of (b) is derived from the structure in (a). In the DCG model, the gray nodes are the observed variables (i.e., the correspondence variables), the white nodes in the plates are the unobserved variables (i.e., the grounding symbols), and the black nodes denote the factors (i.e., representing the conditional relationship between the variables)

(leftmost descendants) of the parent phrase λ_i in the parse tree of the natural language command. Accordingly, the DCG infers the most likely set of planning constraints (in terms of regions) from the language commands.

For example, consider the parse tree of the command "move to the cube" shown in Fig. 2a and the corresponding DCG model shown in Fig. 2b . The child correspondence variable of the phrase "move" is the correspondence variable for the phrase "to." Similarly, the child correspondence variable of the phrase "to" is the correspondence variable for the phrase "the cube". Thus, in this example, each correspondence variable has exactly one child correspondence variable, yielding the inter-plate structure in Fig. 2b. Examining the parse tree also reveals why the factorization in (2) is reasonable: the meaning "move" should be conditionally independent of the noun "cube" given the prepositional phrase describing a location. After all, the correct grounding of the word "move" is an action not depending on whether the target is a cube or a sphere, but depending on the position of the cube.

Finally, the equation in (2) can be factored as (3), where the factor function $\Psi : \Phi \times \Gamma \times \Lambda \times \Phi \times \Upsilon \rightarrow \mathbb{R}$ (e.g., within each plate in Fig. 2b) determines the most likely configuration $\boldsymbol{\phi}^* = \{\phi_{11}, \phi_{12}, \dots\}$ (where each $\phi_{ij} \in \Phi$) given $\gamma_{ij} \in \Gamma^i$, $\lambda_i \in \boldsymbol{\lambda}$, $\Phi_{c_i} \subset \Phi$, and $\Upsilon_{KP} \subset \Upsilon$.

$$\boldsymbol{\phi}^* = \arg\max_{\phi_{ij} \in \Phi} \prod_{i}^{|\boldsymbol{\lambda}|} \prod_{j}^{|\Gamma^i|} \Psi(\phi_{ij}, \gamma_{ij}, \lambda_i, \Phi_{c_i}, \Upsilon_{KP}). \tag{3}$$

In (3), the factor function Ψ is a log-linear model com-

posed of a weighted combination of binary functions, that is,

$$\Psi(.) = \frac{\exp\Big(\sum\limits_{f\in F_{DCG}} \mu_f f(\phi_{ij}, \gamma_{ij}, \lambda_i, \Phi_{c_i}, \Upsilon_{KP})\Big)}{\sum\limits_{\phi_{ij}\in\{-1,0,1\}} \exp\Big(\sum\limits_{f\in F_{DCG}} \mu_f f(\phi_{ij}, \gamma_{ij}, \lambda_i, \Phi_{c_i}, \Upsilon_{KP})\Big)}, \tag{4}$$

where F_{DCG} is the set of hand-coded binary features that evaluate specific traits about a grounding. For example, a linguistic feature can express whether the word "cube" appears in the command $\boldsymbol{\lambda}$, or a geometric feature can identify the spatial characteristics of object aggregations (e.g., whether a region corresponds to the area between two objects). Moreover, each f has a corresponding weight μ_f. In the DCG model, the weights μ_f are learned by maximizing the training set likelihood using a stochastic gradient descent algorithm (i.e., the L-BFGS algorithm[2] [4]).

Note that a limitation of the DCG model is that it assumes the set of symbols (i.e., objects and phrases) are defined a priori so it does not explicitly represent the unknown symbols. Thus, this model is unable to reason about objects and phrases that have not been trained on.

III. Probabilistic Model for Grounding Unknown Symbols

In this section, we extend the DCG model to infer unknown concepts. This is mainly achieved by introducing new symbols to the model. In the subsequent sections, we describe how to infer the grounding 1) unknown phrases or objects, and 2) hypothetical objects that are outside the robot's field of view.

A. Grounding Unknown Phrases or Objects

In this work, a phrase is defined unknown if it has never been a part of a command in the training process. Similarly, an object is considered unknown, if it has never appeared in the world while training the model. The two main steps we take to enable grounding unknown phrases and objects are 1) to introduce a new grounding symbol to explicitly represent an unknown object and 2) to add new feature functions to identify whether an object or a phrase is unknown. Hence, the set of overall groundings can be defined as the union of unknown and known perceived groundings (i.e., $\Gamma = \Gamma_{UP} \cup \Gamma_{KP}$ [3]), and the world model can be represented based on the known and unknown perceived objects as $\Upsilon = \Upsilon_{KP} \cup \Upsilon_{UP}$. The explicit representation of the unknown symbols is illustrated in Fig. 3a.

In the DCG model, the grounding variables are assumed to be conditional independent from each other given the phrases. Similarly, we assume that the known grounding variables are conditionally independent from the unknown grounding variables given the phrases. Note that this is illustrated in Fig. 3a by the absence of edges between the unknown and known symbols. Consequently, by using the new extended sets of groundings and the world model in (3), the factored objective function for the DCG-UPUP model can be written as

$$\boldsymbol{\phi}^* = \underset{\phi_{ij}\in\boldsymbol{\phi}}{\arg\max} \prod_{i}^{|\boldsymbol{\lambda}|} \prod_{j}^{|\Gamma^i_{KP}\cup\Gamma^i_{UP}|} \Psi(\phi_{ij}, \gamma_{ij}, \lambda_i, \Phi_{c_i}, \Upsilon_{KP}\cup\Upsilon_{UP}). \tag{5}$$

In addition to the set of linguistic or geometric feature functions F_{DCG}, in this work, we introduce a new set of binary feature functions, i.e., F_U, which identifies the unknown phrases and objects. For example, the identification of unknown phrases are achieved by keeping a list of known words, and then the corresponding feature f checks whether a phrase in the command is in that list. On the other hand, the detection of unknown objects are realized by quantifying the classification likelihood (e.g., natural entropy [5]) of perceived objects based on the known object (image) classifiers. Accordingly, the factor function $\Psi(.)$ in (5) becomes a log-linear model with the new feature sets as follows:

$$\Psi(.) = \frac{\exp\Big(\sum\limits_{f\in F_{\text{DCG}}\cup F_U} \mu_f f(\phi_{ij}, \gamma_{ij}, \lambda_i, \Phi_{c_i}, \Upsilon_{KP}\cup\Upsilon_{UP})\Big)}{\sum\limits_{\phi_{ij}\in\{0,1\}} \exp\Big(\sum\limits_{f\in F_{\text{DCG}}\cup F_U} \mu_f f(\phi_{ij}, \gamma_{ij}, \lambda_i, \Gamma_{c_{ij}}, \Upsilon_{KP}\cup\Upsilon_{UP})\Big)}, \tag{6}$$

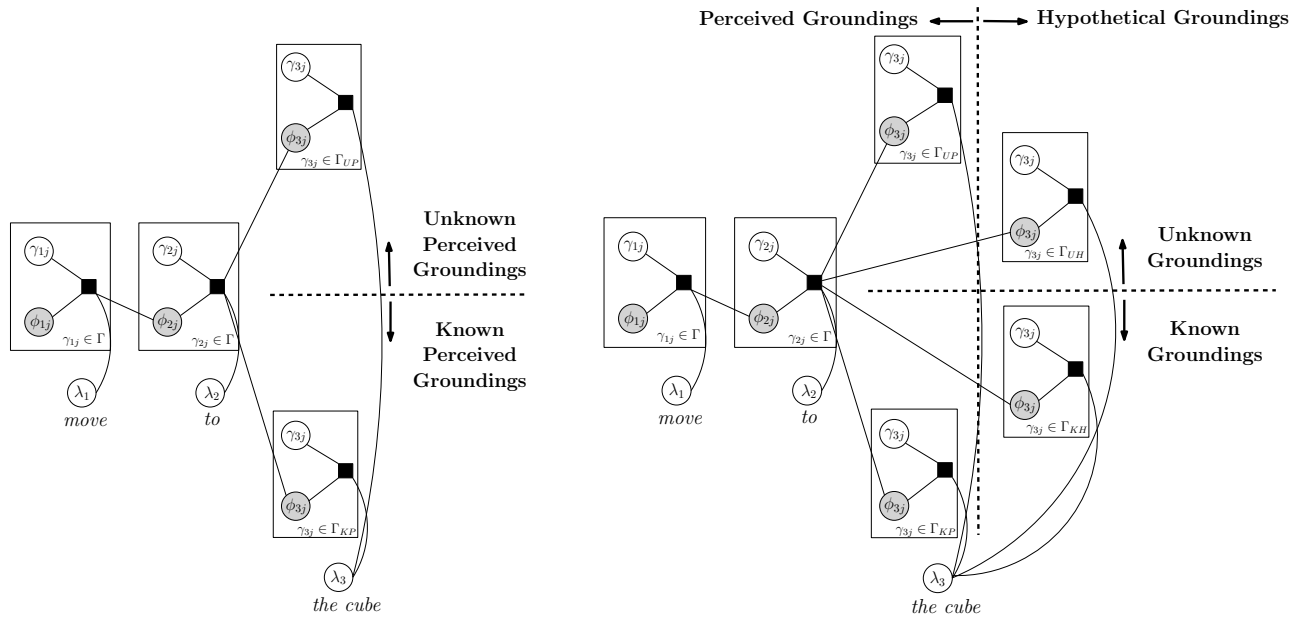
B. Grounding Hypothetical Objects Outside the Field of View

In this work, we define the hypothetical objects as the potential objects that can be located outside the robot's field of view. Now, we extend the DCG-UPUP model to enable grounding hypothetical objects. In existing models, the grounding is achieved based only on the objects that are perceived. However, a robot typically has a limited view of the world, and it is a strong assumption to consider that the given instructions always refer to the objects in the perceived world. To relax this assumption, we propose to add hypothetical objects to the model and let the robot associate a phrase with a hypothesized object when necessary.

Adding hypothetical objects to the model is similar to the process of adding unknown objects to the model. First, after populating a world model by using the sensors of the robot, a single instance of every known object type, as well as one instance of an unknown object, are added to the world model and labeled as hypothetical objects. As a result, the new world model is composed of objects that are known perceived, unknown perceived, known hypothetical, and unknown hypothetical (i.e., $\Upsilon = \Upsilon_{KP}\cup\Upsilon_{UP}\cup\Upsilon_{KH}\cup\Upsilon_{UH}$). Second, new grounding symbols are added to explicitly represent the hypothetical objects. In a similar fashion, the set of groundings are extended as $\Gamma = \Gamma_{KP}\cup\Gamma_{UP}\cup\Gamma_{KH}\cup\Gamma_{UH}$. Third, a new set of binary features (F_H) is introduced to detect whether an object is hypothetical. For example, if the command contains an object that is not perceived based on the current field of view, then the referred object is considered hypothetical. Based on these modifications (the extensions of the world model Υ, the grounding set Γ, and

[2]In our work, we also use the L-BFGS algorithm to learn the weights of the feature functions

[3]In the DCG model, $\Gamma = \Gamma_{KP}$.



(a) The DCG-UPUP model

(b) The DCG-UPUP-Away model

Fig. 3: The graphical models instantiated for the command "move to the cube". (a) The unknown groundings are explicitly represented and the grounding variables are assumed to be perceived. (b) The unknown perceived, known perceived, known hypothetical, and unknown hypothetical groundings are explicitly represented (separated by dashed lines).

the feature functions F), the factored objective function for the DCG-UPUP-Away model can be written as

$$\boldsymbol{\phi}^* = \arg\max_{\phi_{ij} \in \boldsymbol{\phi}} \prod_{i}^{|\boldsymbol{\lambda}|} \prod_{j}^{|\bar{\Gamma}^i|} \Psi(\phi_{ij}, \gamma_{ij}, \lambda_i, \Phi_{c_i}, \bar{\Upsilon}), \tag{7}$$

where $\bar{\Gamma}^i = \Gamma^i_{KP} \cup \Gamma^i_{UP} \cup \Gamma^i_{HP} \cup \Gamma^i_{HU}$, $\bar{\Upsilon} = \Upsilon_{KP} \cup \Upsilon_{UP} \cup \Upsilon_{KH} \cup \Upsilon_{UH}$, and

$$\Psi(.) = \frac{\exp\Big(\sum\limits_{f \in F_{\text{DCG}} \cup F_U \cup F_H} \mu_f f(\phi_{ij}, \gamma_{ij}, \lambda_i, \Gamma_{c_{ij}}, \bar{\Upsilon})\Big)}{\sum\limits_{\phi_{ij} \in \{0,1\}} \exp\Big(\sum\limits_{f \in F_{\text{DCG}} \cup F_U \cup F_H} \mu_f f(\phi_{ij}, \gamma_{ij}, \lambda_i, \Gamma_{c_{ij}}, \bar{\Upsilon})\Big)}. \tag{8}$$

Note that the resulting graphical model for the DCG-UPUP-Away is illustrated in Fig. 3b, where the nouns may ground to 1) known and perceived objects, 2) unknown and perceived objects, 3) known and hypothetical objects, and 4) unknown and hypothetical objects.

IV. Online Learning

In this section, we propose an unsupervised online learning strategy for the model to acquire new symbols. To this end, we present an exploration phase for the robot to search for unknown objects and an incremental learning strategy to learn new symbols.

A. Exploration to Find an Unknown Object

Given a natural language command, a robot can ground the phrases within its world model by solving an inference problem over the DCG-UPUP-Away model. Consequently, a noun phrase can be grounded to a perceived (known or unknown) or a hypothetical (known or unknown) object. In the case of grounding to a hypothetical object, the robot needs to explore its surroundings to find the potential object that is referred by the phrase. There might be several exploration strategies to find the hypothetical object. In this work, we assume that the robot gradually rotates in its current position to change its field of view. As the field of view changes, the world model is updated based on the new perceived objects, and the grounding problem with the same command is solved over the DCG-UPUP-Away model until the noun phrase is grounded to a perceived object.

B. Incremental Unsupervised Learning

Suppose that a robot is given a command with an unknown phrase. The solution over the DCG-UPUP-Away model is the association of the unknown phrase with the unknown object in the environment. When the robot perceives an unknown object as it is initially deployed, then the unknown phrase grounds to that unknown object. If the world model in its initial deployment does not contain an unknown object, it starts to explore the environment (as discussed in the previous section). Whenever it finds an unknown object, then the unknown phrase is grounded to that object.

In addition to grounding unknown phrases to unknown objects, the model is capable of learning new symbols (objects and phrases) based on the past experience. In that case, although a robot starts a mission with a small set of known phrases and objects, it can incrementally increase its knowledge on phrases and objects and perform more efficiently in the future. To achieve this, whenever an unknown

phrase is grounded to an unknown object, we propose an unsupervised learning procedure with the following steps:

Step 1: A new object type is created. For example, if the unknown noun phrase is "apple", a new symbolic apple-type object is created.

Step 2: Based on the given command, the current world model, and the grounding solution, a new training file is created. For example, suppose that the world contains one apple, one cube, and one cone, and the robot initially knows what a cube and cone are. Let the command be "go near the apple". Then, the DCG-UPUP-Away model will correspond to the unknown phrase "apple" with the unknown object apple. After creating the new object type for apple as in step 1, the grounding solution is updated as corresponding the phrase "apple" with the apple-type object. Consequently, the current world model, the given command, and the updated grounding solution constitute a new training file.

Step 3: Since a new object type is created (e.g., apple-type), the set of groundings (Γ) is updated by adding the new symbolic object.

Step 4: Since a new object type is added and an unknown phrase is grounded (from step 3), the set of features are updated as follows: first, a new feature function is created to classify whether an object corresponds to the new type. For example, several images of the apple are taken and an apple classifier is trained. Accordingly, the feature function returns true if an object is likely to be an apple based on the classifier. Second, the feature function to check whether a word is known is updated (e.g., the phrase "apple" is added to the list of known words).

Consequently, after creating a new training file and updating the set of groundings and the feature functions, the log linear model in (8) is retrained to update the weighting functions. Hence, initially unknown symbols become known after the training.

V. Evaluation

The performance of the DCG-UPUP-Away model was demonstrated in two experiments. First, a simulated TurtleBot within randomly generated simulated environments was given a series of user-generated natural language commands. Second, an actual TurtleBot was given specific commands in a laboratory environment in order to demonstrate novel behaviors enabled by the DCG-UPUP-Away model. Both experiments assumed a perfect object recognizer that translates the raw sensor data into symbols into the world model, as well as an initial set of hand-labeled training examples for training the log-linear model to ground cubes, spheres, and cylinders. In all trials, training the model with 55 positive examples took less than 1 minute on a Lenovo Thinkpad X1 Carbon, and grounding a command took under 40 seconds.

A. Corpus

The simulated testing environments were randomly generated in Gazebo. Ten worlds were created, and each was populated with a random collection of objects in randomized locations. In this work, we considered 8 possible object types (including cubes, spheres, and cylinders) in 3 possible colors (red, blue, green), for a total of 24 objects. Each object had a 15% chance of being added to a given map. Using such a procedure to generate environments coupled with the limited field of view of the TurtleBot caused 87% of the objects to be placed outside the initial field of view of the robot, which demonstrates the need for the ability to ground commands to hypothesized objects.

After generating the 10 worlds, the screen shots of a world with a highlighted single object were uploaded to Amazon Mechanical Turk. For each image, the users were instructed to write a command "for approaching the highlighted object." These image-command pairs were saved for evaluating whether a robot, when placed in the corresponding simulated world and given the natural language command, successfully approached the correct object. An example screen shot, with an annotation supplied by a user, is shown in Fig. 4.

B. Testing/Training Setup

Ten image-command pairs were randomly selected without any replacement from the pool of all pairs. Note that one iteration corresponds to a specific pair, and 10 ordered pairs is called a trial. Accordingly, 30 trials were generated, each consisting of 10 iterations, for a total of 300 evaluations. When executing a trial, the TurtleBot was first trained on the initial, hand-curated training set. The robot was then given the natural language command from the first iteration, and then retrained using the initial data supplemented by unsupervised training examples generated by the first iteration. The retrained TurtleBot was given the command from the next iteration, and appropriately retrained after each execution until all 10 iterations have been executed.

C. Perception Pipeline

The perception pipeline involved object detection using the *Edge Boxes* toolbox of Zitnick, Lawrence and Doll'ar [6] in which edges within an image are identified and grouped into candidate objects. Dense SIFT features [7] were extracted from the set of training images (with the VLFeat [8] toolbox) and used to construct a dictionary (of size 600). Spatial histograms over the dictionary of visual words was used to form a feature vector for the images, and the resulting vectors were then classified using a set of one-vs-all support vector

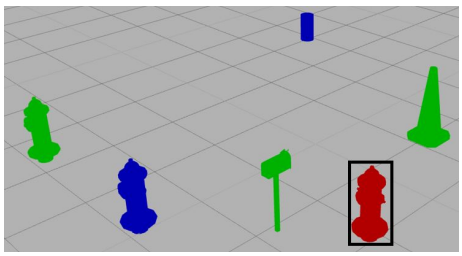


Fig. 4: A simulated world with a highlighted object presented on Amazon Mechanical Turk, labeled by a user as "Move to the red fire hydrant."

machines (χ^2 kernel) [9]. Objects were labeled as "unknown" if none of their computed signed-margins exceeded a *margin threshold parameter*, which we chose to be 0.2.[4]

Images were obtained using the RGB camera from a Microsoft Kinect. We picked a subset of common objects from the standard YCB Object Dataset [10] for evaluation. The system was initialized with knowledge of three different objects (a Cheeze-It box, a Spam tin, and a Coffee can) and with twenty training images per class. When a new object was learned, twenty images were collected (offline) for that class and the classifiers were retrained. Finally, whenever objects were recognized, the new detections were added to the training set, and the classifiers were again retrained.

VI. Results

This section presents the performance results of the DCG-UPUP-Away model based on the grounding accuracy (how likely the model correctly grounds a phrase) and the number of known symbols. Under the grounding accuracy results, we also examine when phrases are grounded to known, unknown, or learned objects.

A. Grounding Accuracy

As discussed previously, the TurtleBot is retrained between the iterations, thus the grounding accuracy may change as a function of iteration number. In fact, the mean grounding accuracy remains between 70% and 90% across all iterations, as shown in Figure 5a. Although the overall grounding accuracy remains relatively constant, the underlying behavior within the DCG-UPUP-Away model changes over the course of a trial. For example, Fig. 5b illustrates 3 curves showing what fraction of correctly grounded phrases refer to known, unknown, or learned objects as a function of iteration number. In the first iteration, nearly 70% of correctly grounded commands refer to known objects, but by the 10^{th} iteration that number has fallen to nearly 10%, replaced almost entirely by correctly grounding to learned objects.

We also tested the performance of the DCG model with the commands containing known and unknown phrases. The DCG model was first trained with a set of commands only including phrases "cube", "sphere", and "cylinder". Then, the trained model was given 100 commands, and 35 of them included known phrases "cube", "sphere", "cylinder" while 65 of them contained unknown phrases "cone", "fire hydrant", "cordless drill", "mailbox", and "door handle". The results, illustrated in Table I, indicate that 30 commands with known phrases were grounded correctly, 5 commands with known phrases were grounded inaccurately, 17 commands with unknown phrases were grounded inaccurately by associating them to wrong objects, and 48 commands containing unknown phrases returned no grounding. This was mainly due to the solutions that could not exceed the selected confidence threshold (i.e., 0.75 in this study). Overall, only 30 commands out of 100 were grounded correctly, and the results demonstrated that the DCG model performs poorly in the case of commands with unknown phrases.

[4]It should be noted that, while perception is not the focus of this work, there is potential for improvement in both the bounding box proposal (e.g. using the depth images) and the classification.

	Accurate	Inaccurate	Uninformed
DCG	0.30	0.22	0.48

TABLE I: The summary of results showing the fraction of accurate, inaccurate, and uninformed groundings of 100 commands.

B. Learned Symbols

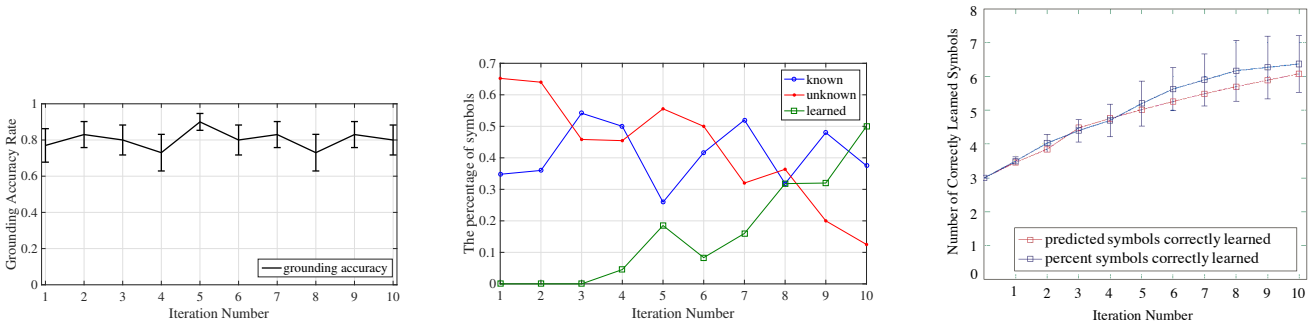
In order to better examine the learning behavior exhibited by the DCG-UPUP-Away model, the other performance metric considered is the number of correctly known symbols. Note that the symbols may be incorrectly learned by associating a phrase with the wrong sort of object due to the nature of unsupervised learning. Initially, the TurtleBot is trained with cubes, spheres, and cylinders, but the generated environments may contain up to 5 additional object types (i.e., fire hydrants, drills, mailboxes, door handles, and traffic cones). Whenever an unknown phrase is grounded to such an unknown object, the TurtleBot learns the new symbol. Thus, one may calculate the expected number of known symbols as a function of the iteration number using combinatorics to count how many unknown objects are present. The recorded number of correctly learned symbols are plotted in Fig. 5c in blue, as well as the expected number in red.

As expected, the blue curve starts at 3 (for the cube, sphere, and cylinder), and stochastically monotonically increases. In 10% of the trials, all 8 symbols were correctly learned. In other trials the DCG-UPUP-Away model incorrectly grounded unknown phrases (and therefore learned an incorrect symbol) or the 10 iterations collectively never referred to the five initially unknown objects, preventing the DCG-UPUP-Away model from ever learning the new symbol. Furthermore, learning symbols correctly improves the grounding accuracy: for each additional correctly learned symbol, the TurtleBot is over 4% more likely to correctly ground a command.

C. Physical Demonstration

In addition to the simulation studies, the DCG-UPUP-Away model was tested on an actual TurtleBot in a laboratory setting. The TurtleBot was placed facing a box (unknown). In addition, a jar (known), a can (known), and fruits (unknown) were located behind the TurtleBot.

Three natural language commands were used to demonstrate various capabilities of the DCG-UPUP-Away model. First, the TurtleBot was given the command "move to the box." The TurtleBot drove to the box, demonstrating that it perceived the box as unknown, recognized the phrase "box" as unknown, and grounded the unknown phrase to the unknown object. Thus, a command was correctly grounded to an unknown perceived object as illustrated in Fig. 7. Second, the TurtleBot was given the command "move to the jar". The TurtleBot rotated in place until the jar came in perception, and then approached the jar. In other words, the command

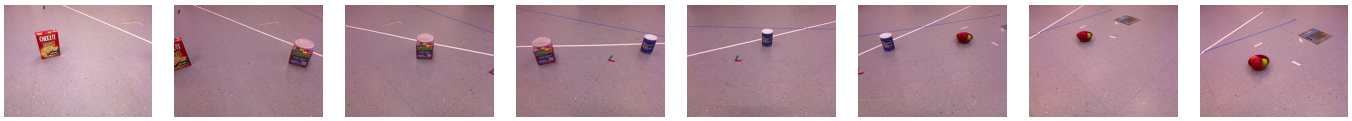


(a) The overall grounding accuracy rate over various iterations.

(b) The mean percentage of known, learned, and unknown symbols during the simulations.

(c) The number of new symbols acquired during the simulations.

Fig. 5: The DCG-UPUP-Away model was used to ground 30 trials, each of which included 10 iterations (i.e., a specific pair of command and world model).



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 6: An illustration of grounding to an unknown hypothetical object. The TurtleBot initially knew all objects in the world other than fruits. The robot was given a command as "move to the fruits". (a) First, it did not see an unknown object in its perceived world so it created a hypothetical unknown object, (b,c,d,e,f,) it explored the world by rotating at its current location until it perceives an unknown object, (g) it perceived an unknown object and grounded to it, (h) it drove to the fruits.

was first grounded to a known hypothesized object, and then it was grounded to a known perceived object once the jar was seen. Finally, the TurtleBot was given the command "move to the fuits". Once again, the TurtleBot explored its surrounding by rotating at its current location and drove to the fruits once it perceived it (as illustrated in Fig. 6).

The experimental results demonstrate two important behaviors: 1) the TurtleBot must have learned what a box was, otherwise the unknown phrase "fruits" would have been grounded to the box, and 2) the TurtleBot grounded the command to an unknown hypothesized object until the box was perceived.

(a) t= 0 sec. (b) t= 45 sec. (c) t= 50 sec.

Fig. 7: An illustration of learning new symbol. The TurtleBot initially did not know what a box is, a command was given as "move to the box". (a) Due to the presence of an unknown object in its perceived world, it grounded the unknown phrase "the box" to the unknown object, (b,c) it drove to the box.

D. Limitations

The previous sections demonstrated that the proposed model DCG-UPUP-Away results in the successful execution of various natural language commands. This section discusses the main limitations of the model. In particular, the most obvious limitation of the DCG-UPUP-Away model is the assumption of referring an unknown phrase to the first perceived unknown object. Moreover, the DCG-UPUP-Away model assumes a one-to-one correspondence between unknown phrases and unknown objects; thus it cannot, for example, learn synonyms by grounding unknown phrases to the known object types.

VII. Related Work

Our work is closely related to solving the grounding problem over probabilistic graphical models that contain three main variables: grounding variables, language command, and the world model. In the previous works, the domains of the grounding variables, the language command, and the world model have been restricted to the known phrases and the perecived groundings [1], [2]. Furthermore, although some works reason about unknown environments, many probabilistic grounding techniques assume fully observable worlds [11], [12].

In this paper, we propose an unsupervised learning process to learn new symbols. An alternative way of grounding unknown or ambiguous symbols can be done via human-robot dialogue. For example, Ros et al. broadly approach resolving language ambiguity using two techniques [13]. First, a robot attempts to model the human's perspective on the scene to determine which objects may be visible to the human. This technique relies on insights from child development studies (e.g., [14]) that show how children employ such reasoning

on their own and has been successfully used in the robotics literature (e.g., [15], [16]). The second strategy relies on the robot asking a human for more information. For example, the robot may ask for spatial relations or object features in distinguishing between objects. Choosing exactly which question to ask, of course, requires reasoning about what information best discriminates among potential groundings. For example, the entropy of the probability distribution over groundings is used to estimate the grounding uncertainty in [17]. Accordingly, higher entropy leads to more questions which improves the grounding accuracy rate. However, a critical issue is the proper balance between too many questions and not enough questions while simultaneously determining what sorts of question to ask (e.g., [18], [19]).

One common approach for autonomous language learning provides a robot with semantic representations of the world that must be associated with language. Such associations may be formally expressed using predicate logic, but ultimately the problem of language acquisition is reframed as a mapping problem from words to pre-defined semantics (e.g., [20], [21], [22], [23]). Unfortunately, hand-labeled representations necessarily require intensive human involvement in generating training data [24]. As a result, some studies considered the opposite approach and try to associate words directly to objects or actions without creating formal symbolic representations. For example, the capability of learning shape categories without being told ahead of time (e.g., four right angles define rectangular objects) was demonstrated by using online raw video data and sentences in [25], [26].

Finally, some studies in the literature considered the idea of hypothesizing objects outside the field of view. For example, Duvallet et al. used a framework to propose a latent map partially observed by the language command [11]. Accordingly, in the case of a ball outside the door, the phrase "pick up the ball outside the door" generates a region of high probability near the door and low probability further away. Sampling from this distribution, as well as updating it as more observations are made, yields a useful map to plan in. Similarly, some other work generated distributions over maps or exactly place objects in unknown environments if their locations are uniquely described [27], [12].

VIII. Conclusion

This paper addressed the problem of understanding natural language commands within a robot's world model. The main contribution of the paper was to propose a new probabilistic graphical model called DCG-UPUP-Away, which allows the explicit representation of 1) unknown phrases or objects, and 2) hypothetical objects that can be outside the field of view. Moreover, the proposed model has the capability to learn new symbols in an online fashion, so the learned phrases or objects become known when they are encountered again. The performance of the proposed model was evaluated via simulations and real experiments, where a TurtleBot was used and various natural language commands were given. The results indicated that the DCG-UPUP-Away model has high grounding accuracy with new symbol acquisition. Some potential future directions can be extending the model to reason about multiple unknown (hypothetical) objects or understanding the synonyms of known objects.

References

[1] S. Tellex, T. Kollar, S. Dickerson, M. Walter, A. Banerjee, S. Teller, and N. Roy, "Understanding natural language commands for robotic navigation and mobile manipulation," in *National Conference on Artificial Intelligence*, 2011.

[2] T. Howard, S. Tellex, and N. Roy, "A natural language planner interface for mobile manipulators," In *International Conference on Robotics and Automation*, June 2014.

[3] R. Paul, J. Arkin, N. Roy, and T. M. Howard, "Efficient grounding of abstract spatial concepts for natural language interaction with robot manipulators," in *Proceedings of Robotics: Science and Systems (RSS)*, Ann Arbor, Michigan, USA, June 2016.

[4] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, vol. 45, no. 1-3, pp. 503–528, 1989.

[5] H. Grimmett, R. Paul, R. Triebel, and I. Posner, "Knowing when we don't know: Introspective classification for mission-critical decision making," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2013, pp. 4531–4538.

[6] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European Conference on Computer Vision*. Springer, 2014, pp. 391–405.

[7] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[8] A. Vedaldi and B. Fulkerson, "Vlfeat: An open and portable library of computer vision algorithms," in *Proceedings of the 18th ACM international conference on Multimedia*. ACM, 2010, pp. 1469–1472.

[9] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of machine learning research*, vol. 9, no. Aug, pp. 1871–1874, 2008.

[10] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *Advanced Robotics (ICAR), 2015 International Conference on*. IEEE, 2015, pp. 510–517.

[11] F. Duvallet, M. Walter, T. Howard, S. Hemachandra, J. H. Oh , S. Teller, N. Roy, and A. T. Stentz , "Inferring maps and behaviors from natural language instructions," in *International Symposium on Experimental Robotics*, June 2014.

[12] T. Williams, R. Cantrell, G. Briggs, P. Schermerhorn, and M. Scheutz, "Grounding natural language references to unvisited and hypothetical locations," 2013.

[13] R. Ros, S. Lemaignan, E. A. Sisbot, R. Alami, J. Steinwender, K. Hamann, and F. Warneken, "Which one? grounding the referent based on efficient human-robot interaction," in *19th International Symposium in Robot and Human Interactive Communication*, Sept 2010, pp. 570–575.

[14] H. Moll and M. Tomasello, "Twelve- and 18-month-old infants follow gaze to spaces behind barriers," In *British Journal of Developmental Psychology*, 2004.

[15] J. G. Trafton, N. L. Cassimatis, M. D. Bugajska, D. P. Brock, F. E. Mintz, and A. C. Schultz, "Enabling effective human-robot interaction using perspective-taking in robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 35, pp. 460–470, 2005.

[16] J. G. Trafton, A. C. Schultz, M. Bugajska, and F. Mintz, "Perspective-taking with robots: experiments and models," in *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005.*, Aug 2005, pp. 580–584.

[17] R. Deits, S. Tellex, P. Thaker, D. Simeonov, T. Kollar, and N. Roy, "Clarifying Commands with Information-Theoretic Human-Robot Dialog," *Journal of Human-Robot Interaction*, vol. 2, no. 2, pp. 58–79, 2013.

[18] T. W. Fong, C. Thorpe, and C. Baur, "Robot, asker of questions," *Robotics and Autonomous Systems*, 2003.

[19] N. Roy, J. Pineau, and S. Thrun, "Spoken dialogue management using probabilistic reasoning," in *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000)*, Hong Kong, 2000.

[20] A. S. Clark, "Unsupervised language acquisition: Theory and practice," 2001.

[21] J. M. Siskind, "Lexical acquisition in the presence of noise and homonymy," in *Proceedings of the 12th National Conference on Artificial Intelligence, Seattle, WA, USA, July 31 - August 4, 1994, Volume 1.*, 1994, pp. 760–766.

[22] J. M. Zelle and R. J. Mooney, "Learning to parse database queries using inductive logic programming," in *AAAI/IAAI*. Portland, OR: AAAI Press/MIT Press, August 1996, pp. 1050–1055.

[23] R. Ge and R. J. Mooney, "A statistical semantic parser that integrates syntax and semantics," in *Proceedings of the Ninth Conference on Computational Natural Language Learning*, ser. CONLL '05, 2005, pp. 9–16.

[24] R. J. Mooney, "Learning to connect language and perception," in *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, ser. AAAI'08, 2008, pp. 1598–1601.

[25] C. Yu and D. H. Ballard, "A multimodal learning interface for grounding spoken language in sensory perceptions," *ACM Trans. Appl. Percept.*, vol. 1, no. 1, pp. 57–80, July 2004.

[26] D. K. Roy and A. P. Pentland, "Learning words from sights and sounds: a computational model," *Cognitive Science*, vol. 26, no. 1, pp. 113–146, 2002.

[27] M. R. Walter, S. Hemachandra, B. Homberg, S. Tellex, and S. Teller, "Learning semantic maps from natural language descriptions," in *Proceedings of Robotics: Science and Systems (RSS)*, Berlin, Germany, June 2013.