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

Mycal Tucker, Derya Aksaray, Rohan Paul, and Nicholas Roy

***Abstract*— This work addresses the symbol grounding problem, that is, understanding the "meaning" of natural language within a robot's workspace. Existing inference techniques typically assume that there is a fixed set of phrases or objects that the robot will encounter during deployment. 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 learning 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. One way of communication is via natural language, which 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 to solve 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. Thus, such methods typically fail to reason about unknown phrases or objects that have never been encountered in the training process. Second, these methods often assume that the location of the object being grounded to is known. In other words, the phrases refer to the 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.

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,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.

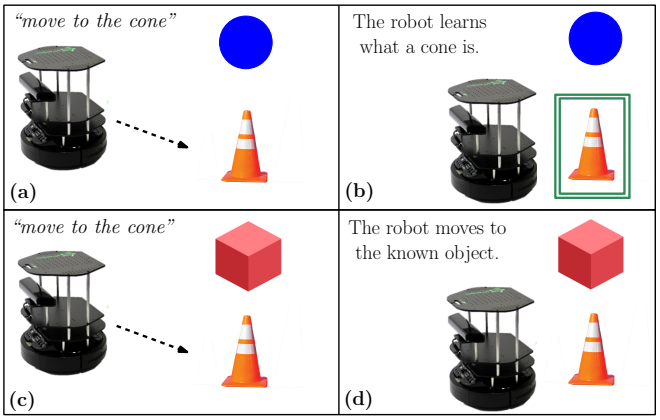


Fig. 1: An illustration of learning and grounding an unknown object. The robot (a) knows what a sphere is, (b) learns what a cone is, (c) sees a cone again, (d) moves towards the cone that is a known object from now on.

This paper proposes a model called the Distributed Correspondence Graph - Unknown Phrase, Unknown Percept - Away (DCG-UPUP-Away), which relaxes two assumptions by 1) explicitly modeling unknown phrases and percepts, and 2) creating hypothetical objects outside the field of view. These two changes yield a model that can ground a large variety of phrases in complex environments, and they 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 robot correctly grounds commands approximately 80% of the time while learning new concepts in an unsupervised manner.

The remainder of this paper is organized as follows: The preliminaries on grounding natural language instructions are introduced in Section II. The technical approach used in developing the DCG-UPUP-Away model is presented in Section III, and online learning of new symbols is presented in Section IV. The model is evaluated in Section V. Existing research in natural language robotics and human-robot interaction that complements this work are reviewed in Section VII. Finally, Sections VIII concludes the paper by summarizing the contributions and future research.

II. Grounding Natural Language Instructions

The work in this paper falls within the field of natural language grounding, which addresses the problem of correctly determining how phrases relate to the real world (e.g., the phrase "go to the cube" means approaching a physical cube). To this end, the general grounding problem can be formulated as a probability maximization problem

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

where $\boldsymbol{\lambda}$ is the natural language command that is a vector of phrases from the set Λ (i.e., the set $\Lambda = \{\text{English phrases}\}$ represents what phrases natural language sentences may be composed of); $|\boldsymbol{\lambda}|$ is the length of the natural language command; Γ is the set of groundings that correspond to semantic notions such as objects, locations, regions, paths, or actions the robot can take and $\boldsymbol{\gamma} \in \Gamma^{|\boldsymbol{\lambda}|}$ is a vector of groundings with a length of $|\boldsymbol{\lambda}|$; and Υ denotes the physical workspace of the robot that aggregates metric and semantic information about the constituent objects. In this formulation, the optimal vector of groundings $\boldsymbol{\gamma}^*$ is the one with maximum likelihood, given a command $\boldsymbol{\lambda}$ and a world model Υ.

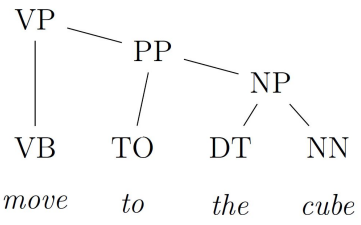
In practice, the domains of Γ, Λ, and Υ in (1) typically include elements from previously seen examples. For example, rather than allowing the set of phrases Λ to include all words in a dictionary, Λ is generally assumed to only contain words that have appeared in the training examples. Moreover, solving (1) is a hard combinatorial optimization problem due to the diversity in language and world.

One way to tackle the complexity of solving (1) is modeling it as an inference over a probabilistic graphical model based on the linguistic structure of the commands. In literature, there exists an efficient model called the Distributed Correspondence Graph (DCG) [2], which discretizes the continuous space of groundings (Γ) as regions and motion constraints and introduces binary correspondence variables (ϕ_{ij}) relating the i^{th} phrase λ_i with the j^{th} grounding variable γ_{ij}. The DCG model assumes the grounding variables as conditionally independent and solves an inference problem as a search over the unknown correspondence variables as follows:

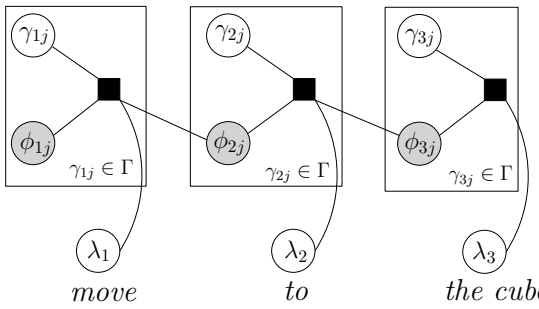
$$\boldsymbol{\phi}^* = \arg\max_{\phi_{ij} \in \boldsymbol{\phi}} \prod_{i}^{|\boldsymbol{\lambda}|} \prod_{j}^{|\Gamma^i|} p(\phi_{ij} | \gamma_{ij}, \lambda_i, \Phi_{c_i}, \Upsilon_{KP}), \tag{2}$$

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 Φ_{c_i} is defined as the set of correspondance variables for the immediate children phrases (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 from the language commands.

For example, consider Fig. 2a that shows the parse tree of a simple command (*"move to the cube"*), and Fig. 2b that shows the corresponding DCG model. 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. After all, the correct grounding of the word "move" is an action that does not depend on whether the target is a cube or a sphere, but it does depend on the position of the cube.



(a) 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.

Finally, the equation in (2) can be factorized 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 composed of a weighted combination of hand-coded binary

[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}).

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 each binary function $f \in F_{DCG}$ belongs to a 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 feature function 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, that is the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm [2].

Note that a limitation of the DCG model is that it assumes a fixed set of symbols (i.e., objects and phrases) 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

This section elaborates how to extend the DCG model to allow 1) grounding unknown phrases or objects, and 2) hypothesizing groundings outside of the robot's field. Then, we present a method to resolve ambiguities via linguistic context, which helps to improve the grounding performance of the proposed model.

A. Grounding Unknown Phrases or Objects

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. As illustrated in Fig. 3a, the unknown symbols are decoupled from the known ones, thus the set of overall groundings becomes the union of unknown and known perceived groundings (i.e., $\Gamma = \Gamma_{UP} \cup \Gamma_{KP}$ [3]). Similarly, the world model can also be decoupled based on the known and unknown perceived objects as $\Upsilon = \Upsilon_{KP} \cup \Upsilon_{UP}$.

Let F_{DCG} be the set of feature functions (e.g., linguistic related or geometric features). In this work, we introduce a new set of binary feature functions, i.e., F_U, which detects the unknown phrases and objects. For example, the detection 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) of perceived objects based on the known object (image) classifiers. Accordingly, by plugging the new extended sets of groundings and the features to (3), the factored objective function for the DCG-UPUP model can be written as

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

where

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

B. Grounding Hypothetical Objects Outside the Field of View

The previous section presented that the DCG-UPUP model can explicitly represent the unknown phrases and objects. However, its performance is limited since the robot can only ground to the perceived objects. As an extension of this model, we introduce the DCG-UPUP-Away, which enables to ground phrases to objects that can be outside the field of view.

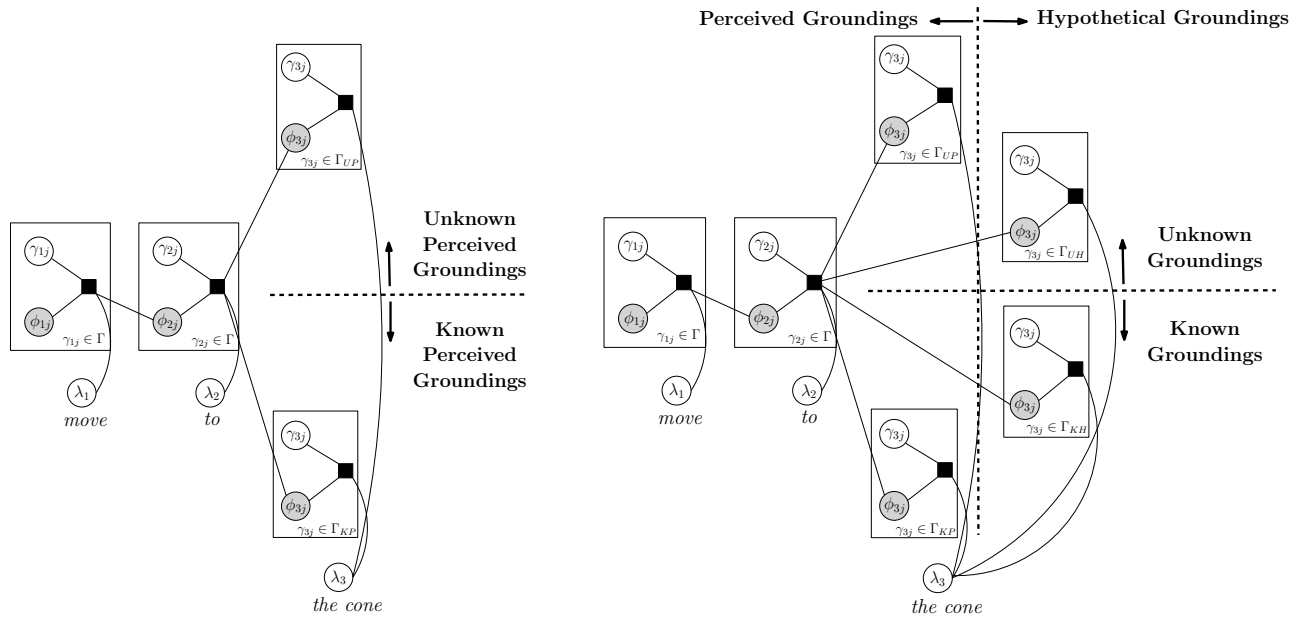
The main process to include hypothetical objects to the model is similar to the process to add unknown objects to the model. First, after populating a world model by using the sensors of the robot, a single instance of every known object type, as well as one instance of an unknown object, are added to the world model and labeled as hypothetical objects. As a result, the new world model constitutes 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 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

[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 constructed for the command "*move to the cone*".

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

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.

C. Resolving Ambiguity via Linguistic Context

One way to improve the grounding performance of the DCG-UPUP-Away model is to allow the association between the natural language adjectives and the object properties. For example, if there exist two cube type objects in the world, one way to distinguish them from each other is to consider their properties such as color or size. In this section, we present how to include color information into the solution of grounding problem over the DCG-UPUP-Away [4]. To this end, a new set of feature functions is introduced, that is $F_C = \{f_{color}, f_{word}\}$ where the feature f_{color} checks the color property of an object and the feature f_{word} detects whether the language command $\boldsymbol{\lambda}$ contains a color adjective. Note that the addition of new features brings only a minor change to (8) where the set of features are extended, i.e, $f \in F_{DCG} \cup F_U \cup F_H \cup F_C$.

[4]Other attributes such as size or shape can be included in a similar way by adding the corresponding feature functions.

IV. Online Learning

A. Exploring the Environment

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 correspondence 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, another contribution of this work is to enable 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 a cone are. Let the given command be "go near the apple". Then, the DCG-UPUP-Away model will correspond 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.

Step 4: The set of features are updated due to adding a new object type and grounding an unknown phrase. To achieve this, a new feature function is created to detect 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. Also, 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. Corpus Generation via User Study

The performance of the DCG-UPUP-Away model is demonstrated in two experiments. First, a simulated turtlebot within randomly generated simulated environments is given a series of user-generated natural language commands. Second, an actual turtlebot is given specific commands in a laboratory environment in order to demonstrate novel behaviors enabled by the DCG-UPUP-Away model. Both experiments assume a perfect object recognizer that translates the raw sensor data into a world model Υ that can be used by the DCG-UPUP-Away model, as well as an initial set of hand-labeled training examples for training the LLM to ground cubes, spheres, and cylinders. In all trials, training the model with 53 positive examples took less than 1 minute on a Lenovo Thinkpad X1 Carbon, and grounding a command took under 40 seconds. The simulated testing environments are randomly generated in Gazebo. Ten worlds are created, and each is populated with a random collection of objects in randomized locations. There are 8 possible object types (including cubes, spheres, and cylinders) in 3 possible colors, for a total of 24 objects. Each object has 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 has 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 screenshots of a world with a highlighted single object are 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 approaches the correct object. An example screenshot, with an annotation supplied by a user, is shown in Fig. 4.

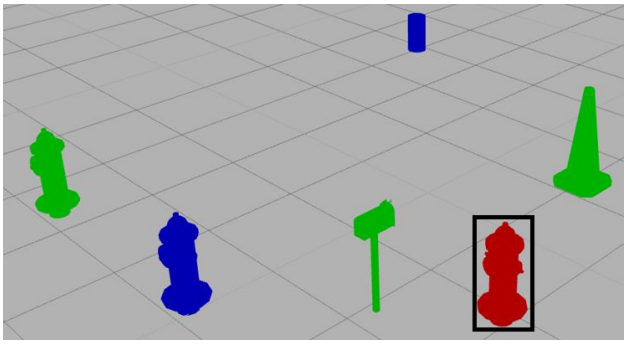
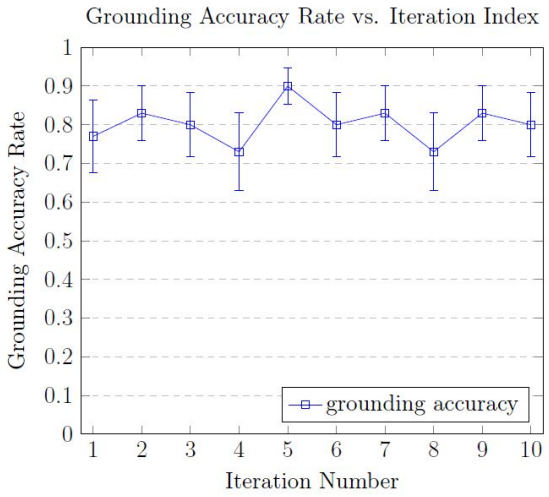


Fig. 4: A simulated world with a highlighted object presented on Amazon Mechanical Turk, labeled by a user as "Move to the red fire hydrant."

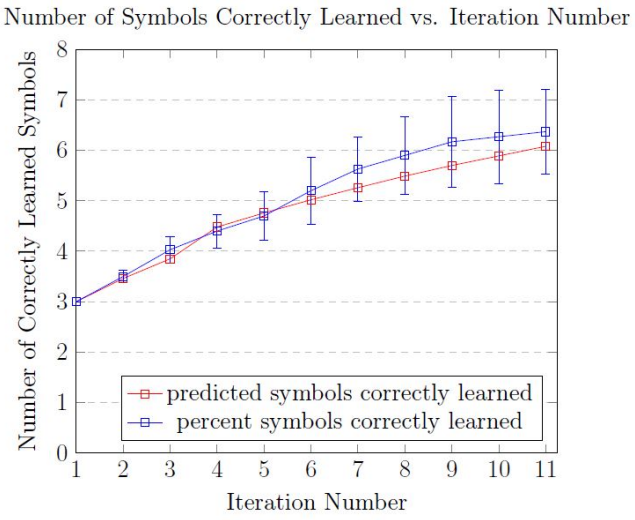
Ten image-command pairs are randomly selected without any replacement from the pool of all pairs. Note that a trial along the paper refers to 10 ordered pairs, and each specific pair is called one iteration. Accordingly, 30 trials are generated, each consisting of 10 iterations, for a total of 300 evaluations. When executing a trial, the turtlebot is first trained on the initial, hand-curated training set. The turtlebot is 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 is given the command from the next iteration, and appropriately retrained after each execution until all 10 iterations have been executed.

VI. Results

The metrics we consider for the performance of the model are the grounding accuracy (how likely the DCG-UPUP-Away model correctly grounds a phrase) and the number of known symbols. We further divide the grounding accuracy results to examine when phrases are grounded to known, unknown, or learned objects.



(a) Overall grounding accuracy.



(b) Number of learned symbols.

Fig. 5: The performance results of the simulation study.

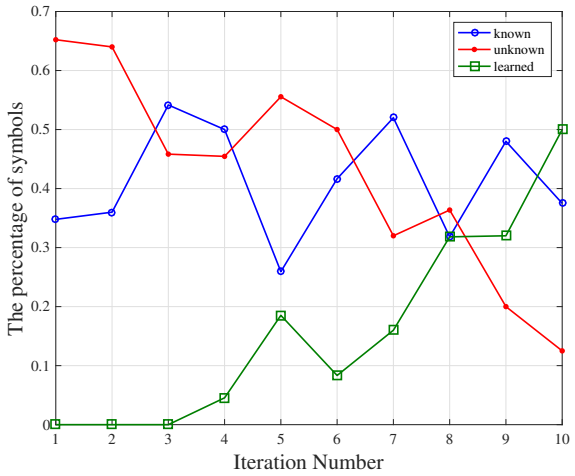


Fig. 6: The percentage of symbols during the simulations.

A. Grounding Accuracy

As discussed previosuly, the turtlebot is retrained between 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. 6 illustrates 3 curves showing what fraction of correctly grounded phrases refer to known objects, unknown objects, 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.

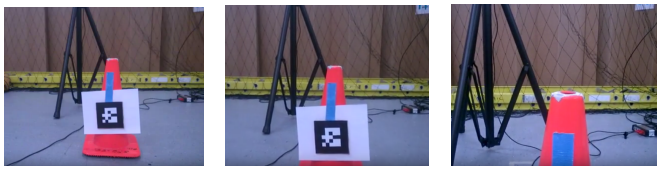
B. Learned Symbols

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. 5b 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 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 cone (unknown). In addition, a cube (known) and a crate (unknown) were located behind the turtlebot. All objects were labeled with the April tags [4], which were used to generate the world model Υ from a kinect camera mounted on the turtlebot.



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

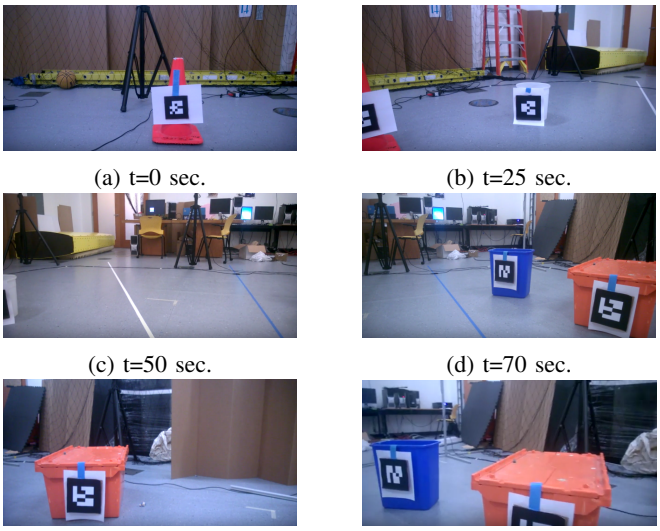
Fig. 7: An illustration of learning new symbol. The turtlebot initially does not know what a cone is, a command is given as "move towards the cone". (a) Since there is an unknown object in its perceived world, it grounds the unknown phrase "cone" to the unknown object, (b,c) it drives to the cone.

Three natural language commands were used to demonstrate all capabilities of the DCG-UPUP-Away model. First, the turtlebot was given the command "move towards the cone." The turtlebot drove to the cone, demonstrating that it perceived the cone as unknown, recognized the phrase "cone" 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 towards the cube." The turtlebot rotated in place until the cube came in perception, and then approached the cube. In other words, the command was first grounded to a known hypothesized object, and then it was grounded to a known perceived object once the cube was seen. Finally, the turtlebot was given the command "move towards the crate." Once again, the turtlebot explored its surrounding by rotating at its current location and drove to the crate once it perceived it (as illustrated in Fig. 8). The experimental results demonstrate two important behaviors: 1) the turtlebot must have learned what a cone was, otherwise the unknown phrase ("crate") would have been grounded to the cone, and 2) the turtlebot grounded the command to an unknown hypothesized object until the crate was perceived. The interested reader is referred to the following link [5] for the videos corresponding to these experiments.

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. One strategy to relax this assumption has been explored in Section III-C by associating language adjectives with object properties. However, a more sophisticated strategy is required for generalizable solutions. 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.

[5] https://www.youtube.com/playlist?list=PL8sYMUToK9s6dAu3qMHHOef8FyhOnDK4E



(a) t=0 sec. (b) t=25 sec. (c) t=50 sec. (d) t=70 sec. (e) t=95 sec. (f) t=120 sec.

Fig. 8: An illustration of grounding to a hypothetical object. The robot initially knows all objects in the world other than a crate. The turtlebot is given a command as "move towards the crate". (a) First, it does not see an unknown object in its perceived world so it creates a hypothetical unknown object, (b,c,d) it explores the world by rotating at its current location until it perceives an unknown object, (e) It perceives an unknown object and grounds to it, (f) it drives to the crate.

VII. Related Works

This 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 [5], [6].

In this paper, we propose an unsupervised learning process to learn new symbols (i.e., new phrases or objects). 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 [7]. 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 that show how children employ such reasoning on their own and has been successfully used in other robotics literature [8], [9], [10], [11]. 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 (e.g., [12]). For example, the entropy

of the probability distribution over groundings is used to estimate the grounding uncertainty in [12]. Accordingly, higher entropy leads to more questions which improves the grounding accuracy rate. Note that a critical issue in robotic question-asking is the proper balance between too many questions and not enough questions while simultaneously determining what sorts of question to ask [13], [14].

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., [15], [16], [17], [18]). Unfortunately, hand-labeled representations necessarily require intensive human involvement in generating training data [19]. As a result, some studies consider the opposite approach and try to associate words directly to objects or actions without creating formal symbolic representations. For example, using online raw video data and sentences, a system is able to learn shape categories without being told ahead of time that four right angles define rectangular objects [20], [21].

Finally, there exist some studies in the literature considering the idea of hypothesizing objects out of perception. For example, Duvallet et al. uses a framework to propose a latent map that is partially observed by the language command [5]. Accordingly, in the example 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 the distribution as more observations are made, yields a useful map to plan in. Similarly, some other works generate distributions over maps or exactly place objects in unknown environments if their locations are uniquely described [22], [6].

VIII. Conclusion

This paper addressed the problem of understanding natural language commands within a robot's symbolic 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 can ground correct objects approximately 80% of the time. 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] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2011, pp. 3400–3407.

[5] 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.

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

[7] 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.

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

[9] ——, "Level 1 perspective-taking at 24 months of age," In *Developmental Science*, 2006.

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.