Learning Unknown Groundings for Natural Language Interaction with Mobile Robots

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

Abstract Our goal is to enable robots to understand or "ground" natural language instructions in the context of its perceived workspace. Contemporary models learn a probabilistic correspondence between input phrases and semantic concepts (or groundings) such as objects, regions or goals for robot motion derived from the robot's world model. Crucially, these models assume a fixed and *a priori* known set of object types as well as phrases and train probable correspondences offline using static language-workspace corpora. Hence, model inference fails when an input command contains unknown phrases or references to novel object types that were not seen during the training. We introduce a probabilistic model that incorporates a notion of unknown groundings and learns a correspondence between an unknown phrase and an unknown object that cannot be classified into known visual categories. Further, we extend the model to "hypothesize" known or unknown object groundings in case a language instruction communicates an interaction with an object present beyond the robot's partial view of its workspace. When the grounding for an instruction is unknown or hypothetical, the robot performs exploratory actions to accrue observations and find the referenced objects beyond the current view. Once unknown groundings are associated with percepts of new object, the model is adapted and trained online using accrued visual-linguistic observations to reflect the new knowledge gained for interpreting future utterances. We evaluate the model quantitatively using a corpus from a user study and report experiments on a mobile platform in a workspace populated with objects from a standardized dataset. A video of the experimental demonstration is available at: `http://groups.csail.mit.edu/rrg/grounding_unknown_concepts`.

1 Introduction

Natural language provides a rich, intuitive, and flexible medium for humans and robots to communicate intent and workspace knowledge while teaming and executing collaborative task. Recent progress has been made in the development of probabilistic models (e.g., [6, 9, 12]) for interpreting or "grounding" natural language instructions in the context of the robot's world representation. These models postulate a space of concepts such as objects, regions, actions etc. derived from the robot's world model, which represent possible meaning conveyed in an input language utterance. The task of language understanding is posed as estimating the likely correspondence between linguistic constituents in the input instruction and the space of concepts

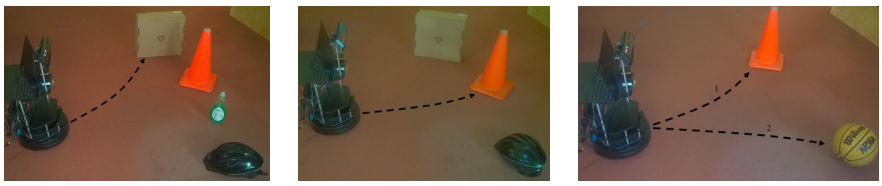
{Mycal Tucker, Derya Aksaray, Rohan Paul, Gregory J. Stein and Nicholas Roy}
Massachusetts Institute of Technology, Cambridge, MA 02139. Authors thank support from the U.S. Army Research Laboratory under the RCTA program and the National Science Foundation.

(or groundings). Current language grounding models rely on a robot's ability to recognize a set of object types (e.g., block, box, chair) in the environment.Further, language grounding models rely on using a predefined training set of linguistic phrases. Crucially, both are assumed fixed and known to the robot *a priori* to the mission. Moreover, it is assumed that the workspace is known with the corresponding object locations, which are also available to the robot ahead of time. However, in realistic workspaces, the types and locations of the objects may only be partially known. Hence, a robot may encounter new phrases that may refer to the objects previously unknown during training and may not be immediately perceptible to the robot. Contemporary grounding models do not possess the ability to ground references in the command that may be unknown or represent object instances that cannot be classified as pre-determined object classes. Hence, they either ignore such cases or yield incorrect groundings when confronted with unfamiliar utterances or unknown percepts.

In this work, we estimate the presence of unknown objects in the scene based on object proposals and a measure of cumulative classification uncertainty given the current set of classifiers. We build on a contemporary language grounding model, the Distributed Correspondence Graph [6], and introduce new probabilistic variables associated with inferred unknown object types populating the workspace. Predictive features based on language and visual cues enable the model to learn a probabilistic association between unfamiliar references in an input utterance and unknown groundings. Further, the descriptive referents in the language utterance are used to infer latent attributes for unknown groundings enabling ambiguity resolution incase of multiple unknown groundings. Once new object instances are encountered, the robot can associate the input linguistic references with percepts and incrementally learn a grounding model that incorporates the newly acquired concept. Further, we extend the formulation to ground commands that may refer to objects beyond the robot's explored workspace. This is accomplished by hypothesizing known and unknown objects beyond the current world model of the robot and taking exploratory actions to seek new observations of the workspace to locate the referenced objects. The model is validated using a corpus of simulated scenarios paired with natural language instructions provided by human subjects from a crowdsourced platform. Results demonstrate high grounding accuracy and the model's ability to learn new object types from visual and linguistic observations through experience and exploration. The proposed model termed the Distributed Correspondence Graph - Unknown Phrase, Unknown Percept - Away (DCG-UPUP-Away) model enables learning the meaning of a large variety of phrases in complex environments and facilitates learning new words and objects in an online manner and contributes towards language-guided models that enable online concept acquisition in complex partially-known workspaces.

2 Background: Grounding Natural Language Instructions

The task of grounding a natural language utterance involves relating each phrase within an utterance to semantic concepts derived from the robot's workspace. For example, grounding the command "navigate towards the red box" involves relating



(a) (b) (c)

Fig. 1: A demonstration of grounding and acquiring unknown concepts referenced in an input utterance. A mobile robot deployed in an environment with *a priori* known (box and helmet) and unknown objects (cone, soap and ball). (a) Given the command "move towards the box," the robot approaches the box by grounding a known phrase to a known object. (b) Receiving the instruction "move towards the cone," the robot drives to the cone object as the phrase "cone" is unknown and the physical cone is perceived as unknown. (c) Given the command "move towards the cone,' the robot follows path 1. When given the command "move towards the ball," the robot follows path 2 towards the ball object after having acquired the concept of a cone previously in (b).

(i) the noun phrase "the red box" to the box object (with red color) in the world model, (ii) the prepositional phrase "towards" with the feasible region adjacent to the box object, and (iii) verb phrase "navigate" as the goal constraint where a robot is close to the box object.

Formally, let Υ denote the world model that includes symbols associated with the known objects populating the workspace. Hence, the world model Υ is a set of tuples, each tuple containing the object type, instance label and a metric location in the environment typically estimated via a perception system. Let Γ denote the set of grounding symbols that correspond to semantic notions such as objects (e.g., blocks, cans, boxes), regions (e.g., near, far, front, top), or constraints (e.g., intersection, distance) that define the goals of actions that a robot can take. The space of grounding symbols is derived as a function of the world model $\Gamma(\Upsilon)$ and cumulatively defines the space of concepts that can potentially be true for the workspace. The process of grounding a language utterance associates each linguistic constituent with likely grounding symbols. Accordingly, the grounding problem can be formulated as:

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

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

One efficient way of solving Eqn. (1) can be achieved via the Distributed Correspondence Graph (DCG) [6]. This model is structured according to the hierarchical structure of the language command. For example, suppose that the language command is "move to the cube". Then, the goal becomes to find the correct associations between "move", "to", "the cube" and objects, regions, constraints 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 and regions 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 groundings Γ includes symbols associated with (i) objects that the robot has been trained with, and (ii) regions and motion constraints that are obtained by discretizing the perceived continuous space. Additionally, the 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} where ϕ_{ij} is the extent to which the language describes or corresponds to the grounding. The main assumption of the DCG model is that the grounding variables are conditionally independent from each other given the phrases. The DCG allows the inference problem to be solved as a search over the correspondence variables as follows:

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

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 [1] of λ_i, 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 (leftmost descendants) of the parent phrase λ_i in the parse tree of the natural language command. Inference on the DCG factor graph yields the most likely set of planning constraints (in terms of regions) from the language commands. Note that the parse tree also reveals why the factorization in Eqn. (2) is reasonable: since 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, the meaning "move" should be conditionally independent of the noun "cube" given the prepositional phrase describing a location. Finally, Eqn. (2) can be factored as in Eqn. (3), where the factor function $\Psi : \Phi \times \Gamma \times \Lambda \times \Phi \times \Upsilon \to \mathbb{R}$ (e.g., within each plate in Fig. 2b) determines the most likely configuration $\phi^* = \{\phi_{11}, \phi_{12}, \dots\}$ (where each $\phi_{ij} \in \Phi$) given $\gamma_{ij} \in \Gamma^i$, $\lambda_i \in \lambda$, $\Phi_{c_i} \subset \Phi$, and $\Upsilon_{KP} \subset \Upsilon$.

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

The factor function Ψ is a log-linear model composed of a weighted combination of binary functions, that is,

[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 prepositional phrase (e.g., "front of a box") then Γ^i contains symbols denoting the discretized spatial regions (i.e., front, behind, left etc.) with respect to the objects under consideration (i.e., Γ^{RO}). If λ_i is a verb phrase referring to the actions that the robot can take (e.g., "move towards a box", "pick up the block"), then Γ^i contains the set of constraints defined with respect to pairs of regions (e.g., picking a block can be implicitly expressed as a intersection constraint between the robot's end-effector and the region occupied by the object).

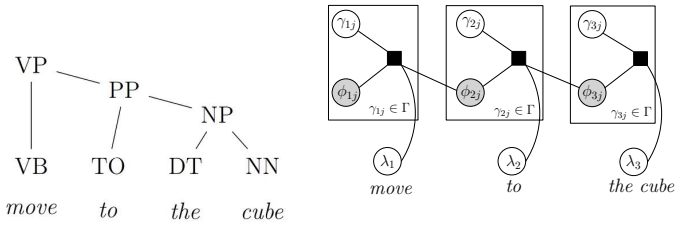


Fig. 2: The parse tree (left) and the corresponding DCG factor graph (right) instantiated for the instruction "move to the cube". The factor graph is structured as per the dependency relations in the parse tree. In (b) the gray nodes are the unobserved variables (i.e., the correspondence variables), the white nodes in the plates are the observed variables (i.e., the grounding symbols), and the black nodes denote the factors (i.e., representing the conditional relationship between the variables)

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

where F_{DCG} is the set of hand-designed binary features that evaluate specific traits about a grounding. For example, a linguistic feature can express whether the word "cube" appears in the command λ, 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. Feature weights are learned via maximizing data log-likelihood.

A limitation of the DCG model is that it assumes a predefined set of object type symbols and linguistic phrases. Thus, this model is unable to reason about objects and phrases that it has not been trained on. Further, the model assumes that the world model is complete, in that all objects to be considered in the grounding process are known. Hence, it is unable to interpret an instruction that references an object beyond the limited perceptual view of the robot.

3 Proposed Probabilistic Model

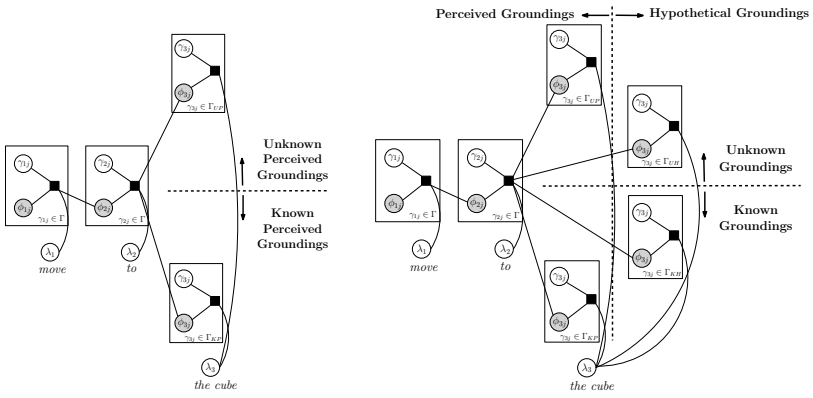
In this section, we present a probabilistic model that introduces the notion of unknown grounding symbols in the DCG formulation. Further, we endow the model with features based on linguistic and perceptual cues to ground unfamiliar phrases to perceptually unknown objects. Moreover, we address the issue of grounding to objects beyond the field of view of the robot in the next section.

3.1 Grounding Unknown Phrases or Objects

A phrase within a language utterance is considered "unknown" if the robot does not possess a model for associating a meaning or a "grounding" for it. This is the case when the phrase has not been encountered previously (during training or operation) and does not possess an associated grounding label. Hence, the robot lacks the ability to predict its meaning. Here, we restrict the space of unknown phrases to only

include unknown noun phrases indicative of unknown object types in the robot's world models. We assume that the robot has a perception system that yields object detections or "proposals" as well as a classification model that yields a distribution over class labels for a pre-defined set of classes. An object proposal is determined to be known if the posterior over class labels given observations is peaked, and it is labeled as "unknown" if the distribution is significantly uninformed. In this work, we rely on a visual sensor for perception of objects in the workspace. We perform object detection using the *Edge Boxes* toolbox of Zitnick, Lawrence and Dollar [14] in which the edges within an image are identified and grouped into candidate objects. Dense SIFT features [1] are extracted from the set of training images (using VLFeat [13]) and used to construct a dictionary (of size 600). Spatial histograms over the dictionary of visual words were 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 machines (χ^2 kernel) [4]. Objects were labeled as "unknown" if none of their computed signed-margins exceeded a margin threshold m_{thresh} parameter. Since, classification margins shrink with growth in the number of classes, the threshold is normalized by the number of categories presently in the model exponentiated by a parameter Z_{exmp} as: $\frac{m_{th}}{(n-1)^{Z_{exp}}}$. The parameters m_{th} and Z_{exp} were estimated as 0.6 and 1.2 respectively through cross-validation using a training dataset.

In this work, we introduce new probabilistic variables for unknown objects in the model as well as predictive features to enable learning to ground unknown phrases with unknown objects. The set of groundings can be expressed as the union of unknown and known perceived groundings (i.e., $\Gamma = \Gamma_{UP} \cup \Gamma_{KP}$). Note that in the DCG model, $\Gamma = \Gamma_{KP}$. The world model can be represented based on the known and unknown perceived objects as $\Upsilon = \Upsilon_{KP} \cup \Upsilon_{UP}$. Note that Υ_{UP} is composed of an



(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). Note that the proposed model introduces unknown and hypothetical grounding symbols in the DCG formulation. The baseline DCG model only possesses grounding symbols for known perceived entities.

unknown symbolic object in view and serves as an abstraction for an object type other than the known object types. Figure 3a illustrates the graphical model for the proposed model.

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 Eqn. (3), the factored objective function for the DCG-UPUP model can be written as

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

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 (F_U) which predict grounding for unknown phrases and objects. For example, the identification of unknown phrases is 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. Also, a feature is introduced to distinguish known grounding symbol types from the unknown ones. The factor function $\Psi(.)$ in Eqn. (5) becomes a log-linear model with the new feature sets as:

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

3.2 Resolving Ambiguity

The model described above enables grounding for unfamiliar phrases to unknown object types. However, in case of multiple unknown objects in the scene, the grounding for an unfamiliar phrase would result in multiple equi-probable unknown objects. Hence, the model leverages information from additional linguistic context in the utterance (e.g., adjectives, part-of-speech tags) and perceptual cues (e.g., color, size) to infer object attributes that inform the posterior distribution over probable groundings. In order to infer attributes from language and visual observations, the model incorporates features that determine the presence of adjectives in the input phrases as well as express statistics related to the perceived colors (e.g., using HSV values). The feature associations are learned from training data enabling the model to predict attribute types based on the visual and linguistic observations. In case the groundings are uninformed or the cues are either absent or less differentiating, the model must query the human for disambiguation [2].

3.3 Grounding Hypothetical Objects Outside the Field-of-View

In the models described so far, 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 an overly strong assumption that the given instruction always refers to the objects in the perceived world. To relax this assumption, we propose a model that hypothesizes objects and let the robot associate a phrase with a hypothesized object when necessary. In this work, we define the hypothetical objects as potential objects that may be located outside the robot's field of view, and we extend the DCG-UPUP model to enable grounding hypothetical objects.

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 symbolic objects that are known perceived, unknown perceived, known hypothetical, and unknown hypothetical (i.e., $\Upsilon = \Upsilon_{KP} \cup \Upsilon_{UP} \cup \Upsilon_{KH} \cup \Upsilon_{UH}$). Note that Υ_{KH} is composed of symbolic objects that are known but not in perception so those are mapped to known hypothetical objects. On the other hand, Υ_{UH} is composed of a symbolic unknown object that is not in the current field of view and represents any object other than the set of known objects. Second, new grounding symbols are added to explicitly represent the hypothetical objects. In a similar fashion, the set of groundings are extended as $\Gamma = \Gamma_{KP} \cup \Gamma_{UP} \cup \Gamma_{KH} \cup \Gamma_{UH}$ where Γ_{UH} contains the symbolic unknown hypothetical objects. 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:

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

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

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

Note that the resulting graphical model for the DCG-UPUP-Away is illustrated in Fig. 3b, where the nouns may ground to (i) known and perceived objects, (ii) unknown and perceived objects, (iii) known and hypothetical objects, and (iv) unknown and hypothetical objects. Moreover, the model is trained to find the weights in Eqn. (8) based on the given perceived world and a corpus of commands referring to objects in perceptual view and outside the field of view.

3.4 Taking Actions and Learning New Grounding Symbols

Actions based on inferred groundings. Given a natural language command and the current world model, the robot estimates the most likely groundings using the DCG-UPUP-Away model. The estimated grounding symbols can correspond to the set of (i) known or unknown groundings emanating from the perceived world model or (ii) hypothetical known or unknown grounding symbols. If the estimated grounding corresponds to a symbol for a known object in perceptual view, the robot can directly plan and execute a trajectory to satisfy the human's intent. If the estimated grounding corresponds to a symbol for a known object outside the field of view (i.e., hypothetical known object), the robot initiates an exploratory motion to acquire new observations of the environment and re-estimates groundings when a coherent set of detections indicative of newly perceived objects are obtained. Specifically, if the newly encountered object belongs to a known object class, the model re-grounding step associates the previously hypothetical grounding symbol to the encountered object and plans a trajectory to reach the object of interest. In case the estimated grounding is an unknown grounding within perceptual view the robot executes the intended motion in relation to the inferred object. Further, the model acquires the new object type and performs online re-training, a process we detail subsequently. Finally, if the inferred grounding is of type hypothetical unknown, the robot continues to explore and detect objects in the environment till an object is classified as unknown or if the exploration fully covers the extent of the workspace. In the former case, the model performs a data association step between the hypothetical unknown grounding symbol and the perceived unknown object and performs the intended motion. In the latter, the grounding remains unknown even after fully exploring the environment.

Acquiring and learning grounding symbols. Our goal is to enable the robot to begin operation with a small set of known phrases and objects and incrementally increase its knowledge about concepts in the workspace through experience. The grounding for a novel phrase in an input language command to unknown object classes is indicative of new information about a previously unknown object type populating the workspace. The model acquires the new symbol by expanding the space of known grounding symbols and updates the model through online re-training using the newly acquired observations enabling learning of the new object types for future inferences. The noun phrase in the input language utterance is used to instantiate a new object type symbol. The space of grounding symbols Γ is expanded to include the new object type. A set of new observations are acquired for the unknown object and are used to instantiate and train a new classification model. In order to learn to ground commands that reference the newly acquired object symbol, the feature sets employed in the log-linear model are expanded. Additional features include feature that determine the presence of the lexical token in the input phrases and the occurrence of the new object type as determined from the perception system. Consequently, after acquiring the new training datum and updating the set of groundings and the feature functions, the log linear model in Eqn. (8) is re-trained to update the feature weights. The incremental method outlined above enables the model to incorporate a new object type grounding symbols and allows the the model to infer a correspondence between an input instruction and the acquired symbol type in subsequent inferences.

Note that the learning of new grounding symbols is *myopic*, since an unknown grounding for the current phrase is used for model re-training. It is possible that the grounding for an unfamiliar phrase with an unknown perceived object may be rendered less probable with new evidence from future observations (visual or language utterances). Hence, there is scope for incorporating *non-myopic* approaches (such as branch and bound trees) that allow symbol acquisition decisions based evidence over longer time horizons. Further, the model employs a pre-defined set of exploratory behaviors (e.g. rotating the camera for gathering views) for grounding instructions that refer to out-of-view groundings. Learning informative exploration policies from experience remains part of future research.

4 Evaluation

The proposed model was evaluated in two sets of experiments. Quantitative evaluation was carried out using a simulation environment with the robot performing a series of navigation tasks in randomly generated scenes based on user-generated natural language commands. Further, the model was used to command a TurtleBot mobile platform operating in a laboratory environment populated with objects from a standardized data set. We discuss model training in the next section followed by evaluation details and results.

4.1 Model Training and Inference

The language understanding model was trained using an aligned corpus consisting of 55 language instructions paired with a world configuration resulting in 1.15×10^6 training examples for constituent grounding factors. A total of 2160 features were used for training. Parameters were trained by maximizing data likelihood using a quasi-Newton optimization method [7]. Average training time was found to be 34.97 ± 0.12 seconds on an i7 quad-core machine with Ubuntu 14.04 system using a multi-threaded implementation (with 5 threads).

The inference procedure begins by estimating a world model from an observed image from high-confidence object detections and recognition using pre-trained classifiers (for an initial set of object categories). The estimated world model enables instantiating the space of grounding symbols within which the input command is interpreted. The input instruction is parsed using a generative phrase-grammar and informs the structure of the instantiated factor graph for the DCG-UPUP-Away model. The set of true correspondences (indicative of expressed groundings) are determined using a beam search procedure (with beam width 4) that incrementally evaluates candidate grounding solutions from child phrases to parent phrases in the factor graph model. Average inference runtime was found to be 3.32 ± 0.24 seconds in our experiments. Conditioned on the estimated groundings, a deterministic procedure initiated an exploratory behavior in case the groundings were hypothesized and out of view or acquired observations and initiated re-training if the grounding was unknown.

4.2 Quantitative Evaluation using a Corpus from User Study

Simulated environment and language corpus. A set of 10 simulated scenes were generated by randomly sampling objects from 8 possible object types (cubes, spheres, cylinders, cones, fire hydrants, door handles, mailboxes and power drills) each possessing 3 colors (red, blue, green). Each object was picked with 15% chance of being added to a scene. The location of the object and the robot was randomized. The simulated robot did not know which particular objects were in vicinity and used a simulated camera to detect objects in view. The simulated robot did not know which particular objects were in vicinity and used a simulated camera to detect objects in view. Approximately, 87% of the objects to be placed outside the initial field-of-view of the robot. The simulated workspaces were used to collect a corpus of language instructions from human subject via the Amazon Mechanical Turk platform. Subjects were presented with a full view of the simulated environment with a randomly sampled object indicated as the goal location. Subjects were requested to provide language input to command the robot towards a indicated object in the scene, which could include objects present beyond the robot's initial view. A total of 390 varied natural language instructions (paired with simulated scenes) were obtained, e.g., "navigate to the green door handle", "go near the red traffic safety cone", "move behind the fire hydrant" etc.

Experimental setup. The language grounding model was initialized and trained with a randomly sampled subset of the full corpus. The initial training set included language commands that referred to three object types: cubes, spheres, and cylinders. A set of views of the sampled object types were used to pre-train the perception system. Note that the generated environments were populated with up to an additional 5 object types: fire hydrants, drills, mailboxes, door handles, and traffic cones. Pre-training with a limited set of object categories and a limited field of view of the robot't sensor ensured a rich set of test cases for grounding instructions to unknown, hypothetical (known and unknown) symbols. Next, a set of 30 evaluation trials were constructed where each trial consisted of a randomly sampled series of 10 natural language commands. Each command was paired with the associated environment and total of 300 instructions were used. Within each trial iteration, if a grounding was estimated to be unknown, new observations were collected and the model was re-trained. The ground truth groundings were assigned by hand. Whenever an unknown phrase is grounded to an unknown object, the model learned the new symbol.

Acquiring grounding symbols across trials. Figure 4a illustrates the number of correctly acquired grounding symbols for the proposed learned model (shown in blue) averaged for 30 trials with 10 iterations each. An object type is considered correctly acquired if the unknown phrase is grounded correctly to an unknown object (or a single correct grounding in case there are multiple unknown hypotheses). In the first iteration, the model was initialized with 3 (for the cube, sphere, and cylinder) object types which increased monotonically during the experiment. All trials succeeded in acquiring at least 6 symbols and 10% trials acquired the full set of 8 symbols. Note that the average number of known symbols as a function of the iteration number can be determined analytically based on the sampling fraction used to generate the simulated worlds and the number of unknown symbols that appear in each iteration. The analytically determined expected number of unknown symbols

that can be acquired in each iteration is plotted (in red). The analytic values represent the best case symbol acquisition performance over iterations averaged over trials. The estimated average number of acquired symbols (in blue) closely followed (and marginally exceeded) the expected maximum symbols that can be acquired in each iteration (in red). The final trend line (in green) plots the number of learned symbols for the model with no incremental learning is performed. In the absence of learning new symbols (the baseline DCG model), the number of learned symbols remain static at 3 as indicated by the horizontal line. Errors in acquiring a correct symbol occurred primarily in scenarios with multiple unknown grounding candidates where inferred attributes from lexical cues were insufficient in resolving the correct grounding with high confidence. The variance in the number of acquired symbols increases over time since longer time horizons allow higher probability for errors during autonomous symbol learning.

Grounding accuracy across trials. Next, we evaluate the average grounding accuracy for the corpus. An instruction was considered correctly grounded if the likelihood of the inferred grounding set exceeded a confidence threshold (0.75). Fig. 4b plots the grounding accuracy across iterations averaged over 30 trials. The mean grounding accuracy was found to be high and ranged between 70% and 90% across iterations. Figure 4c details the overall grounding accuracy (in Fig. 4b) and plots the composition of instructions correctly grounded as those referencing unknown and learned concepts. Initially, 65.2% of the correctly grounded instructions referred to unknown groundings which decreased to 12.5% at the end of the experiment. On the other hand, the fraction of correctly grounded instructions that referenced learned object types increase from zero in the first iteration and rise till 50.1% at the end of the experiment. As shown in Fig. 4a, the number of acquired symbols monotonically increase with iterations. Note that even after symbol acquisition, the online training may be insufficient for accurately grounding all future instructions with high confidence. This type of errors is evident in iterations 5 and 6 in Fig. 4c where monotonic increasing trends are violated momentarily and the observed grounding accuracy fraction was lower even though the number of acquired symbols increased monotonically. Figure 4d compares the cumulative grounding accuracy with the trained object recognition system in comparison with manual or ground truth object labels (average values over 10 trials with 5 iterations each). The language grounding module remained the same for both evaluations. The overall grounding performance ranged between 0.6 and 0.9, and was obtained to be marginally lower compared to the grounding accuracy when manual labeling was used. Figure 5 presents representative examples of resolving ambiguity in case of multiple unknown groundings.

Baseline DCG. The DCG model [6] was also used with the commands containing known and unknown phrases as a baseline. Note that the DCG model lacks the notions of unknown groundings. The 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 indicate that 30 commands with known phrases were grounded correctly, 5 commands with known phrases and 17 commands with unknown phrases were grounded inaccurately by associating them to wrong objects, and 48 commands containing unknown phrases possessed no

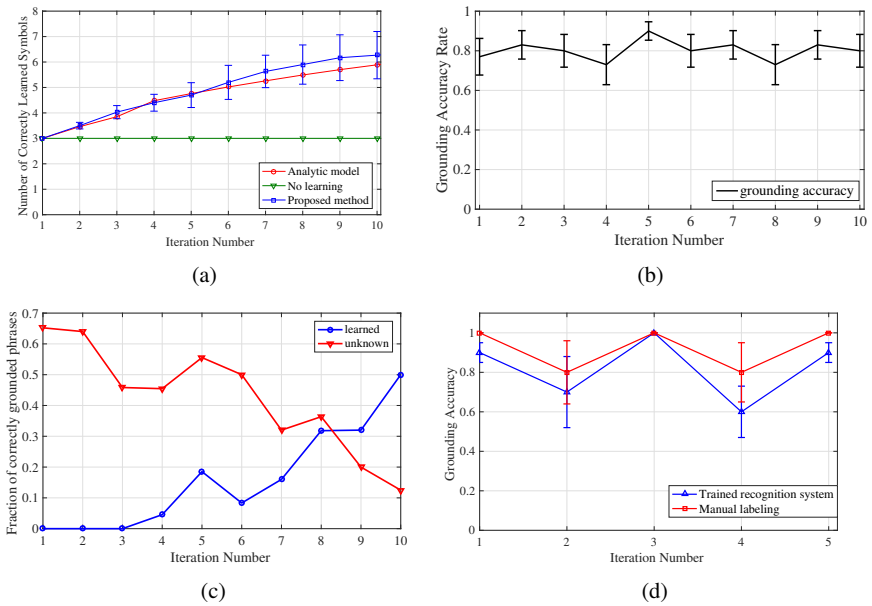


Fig. 4: (a) Average number of new symbols acquired with the DCG-UPUP-Away model (blue), mean number of predicted symbols computed analytically (in red) and number of symbols for the model with no incremental learning (in green). (b) Grounding accuracy for the DCG-UPUP-Away model averaged over 30 trials with 10 iterations each. (c) The mean percentage of correctly grounded phrases that referenced unknown and learned groundings. (d) Grounding accuracy with the trained object recognition system vs. manual labeling. Accuracy values averaged over 10 trials and 5 iteration each. Accuracy for the recognition system were found to be marginally lower. A smaller number of trials in (d) compared to (b) were due to the larger runtime required for running the experiments. Hence, trials and total number of iterations were sub-sampled.

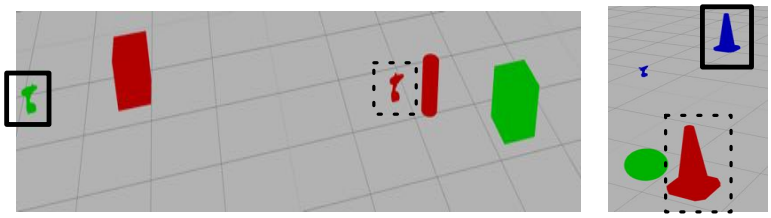


Fig. 5: Ambiguity resolution. Simulated robot-centric views shown above. (Left) The language grounding model has acquired the cylinder and the cube object types. The robot receives the command "got to the green cordless drill" where the drill object type is unknown. The most likely unknown grounding (with a continuous black outline) is inferred using attribute cues in the utterance. The alternative hypothesis (outlined with a dotted line) is less probable. (Right) The model possesses the sphere (green) and drill (blue) object types. The robot receives the command "move to the blue traffic safety cone". The object type cone is unknown and is grounded to the correct object (with a black outline) using color attribute inferred from the command.

expressed groundings. This was mainly due to the solutions that could not exceed the selected confidence threshold. 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.

4.3 Physical Demonstration on a Mobile Robot

The proposed grounding model was deployed on the TurtleBot mobile platform. The video demonstration of the experiments has been submitted as supplementary material along with this manuscript and can be found from the following link: `http://groups.csail.mit.edu/rrg/grounding_unknown_concepts`.

The system was initialized with knowledge of three different objects (a jar of cubes, 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. The robot was equipped with a symbol grounding model trained with the jar and the can as known concepts. The robot was positioned facing the box object (unknown). The remaining set of objects a jar (known), a can (known), and a bowl of fruits (unknown) were placed in the workspace radially from the robot and about the same distance as the box object such that they were outside the starting field of view of the robot as shown in Fig. 6 (left).

Figure 6 (right) demonstrates the trial where the robot is instructed to "move to the box". The robot's world model is populated with a known object jar and an unknown object since the robot does not possess a classification model for recognizing a box. The inferred grounding for the input phrase "the box" is unknown and associated with the only unknown object within the robot's world model. The robot then plans a trajectory to approach the inferred object using visual servoing. The input language utterance paired with the visual observations of the novel object are used to incrementally adapt and re-train the symbol grounding model and the set of classifiers allowing the robot to learn and persist the new object grounding.

Next, the robot was instructed to "move to the jar". The robot performed an exploratory motion by rotating its view-point in place until a jar came in view, and then approached the jar. Overall, the command was first grounded to a known hypothesized object, and then it was grounded to a known perceived object when the jar came into the robot's view. Note that re-grounding was performed when a coherent set of detections were obtained. Finally, the robot was given the command "move to the fruits". Once again, the TurtleBot explored its surrounding by rotating at its current location and drove to the fruits once the object came into the field of view (Fig. 7).

5 Related Work

Contemporary language grounding models [6, 9] assume that the space of linguistic phrases and object types in the environment is known. Further, most probabilistic

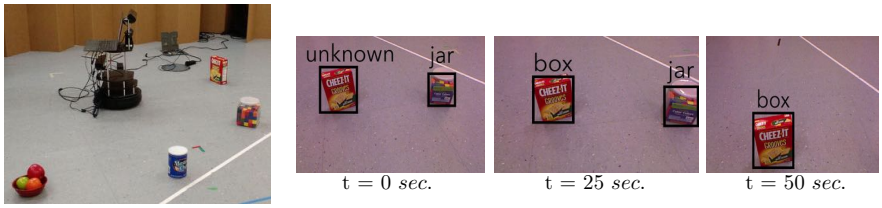
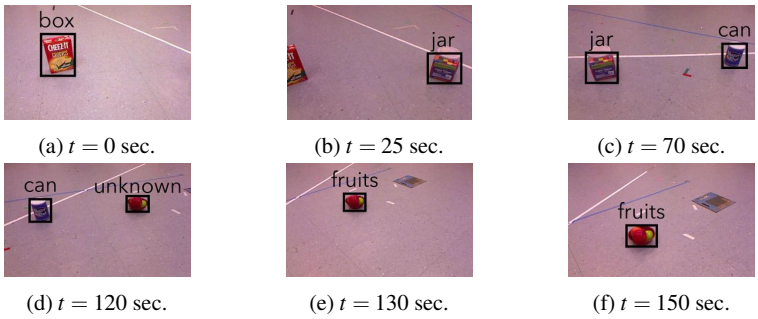


Fig. 6: (Left) Demonstration scenario with the TurtleBot mobile platform in an environment populated with objects (i.e., box, jar, can and fruits in clockwise order) from the YCB data set. The robot is equipped with Kinect sensor with a limited field of view ($62^\circ \times 48.6^\circ$). (Right) Acquiring a new grounding symbol. The robot possessed a model for grounding a jar object but was not trained to recognize or ground a box object. The robot receives a command "move to the box". Due to the presence of an unknown object in its perceived world, the model grounded the unknown phrase "the box" to the unknown object and drove to the box. Online re-trained was performed with the acquired set of visual observations and lexical token "the box". Inference in 2.34 seconds.



(a) $t = 0$ sec. (b) $t = 25$ sec. (c) $t = 70$ sec.

(d) $t = 120$ sec. (e) $t = 130$ sec. (f) $t = 150$ sec.

Fig. 7: Grounding an unknown hypothetical object. The robot possessed knowledge about all objects other than the fruits, and received the command "move to the fruits". (a) The robot did not see an unknown object in its perceived world so it created a hypothetical unknown object. (b,c) It explored the world by rotating at its current location. (d,e) It perceived an unknown object and grounded to it. (f) It drove to the fruits. The robot acquired knowledge about the fruit object from the new observations. Time instants indicated below each frame. Inference in 2.17 seconds.

grounding techniques assume fully observable worlds [3]. Alternatively, the grounding for unknown or ambiguous symbols can be accomplished via dialogue. Knepper et al. posed targeted queries to the human partner using inverse semantics [10]. Selecting which question to ask requires reasoning about what information best discriminates among potential groundings. For example, the entropy of the probability distribution over groundings was used to estimate the grounding uncertainty in [2]. However, determining the quantity and the type of questions becomes a critical issue for acceptable grounding performance (e.g., [5, 11]). Duvallet et al. [3] used the knowledge of object types inferred from language to hypothesize objects beyond the field of view of the robot to guide a semantic mapping algorithm. However, the model assumes that the object types and linguistic references are known and hence ignores reasoning about unknown groundings. Finally, reasoning about unmodelled concepts

was also studied by Nyga and Beetz [8], where they used semantic similarity to known concepts in a Markov logic network.

6 Conclusion

In this work, we introduced a probabilistic graphical model, DCG-UPUP-Away, that allows the explicit representation of (i) unknown phrases or objects, and (ii) hypothetical objects that can be outside the field of view. Further, the model can acquire new grounding symbols in an online manner, so the learned phrases or objects become known when they are encountered again. The proposed model was evaluated via simulations and real experiments with a Turtlebot mobile robot. Results demonstrated that the proposed model displays a high grounding accuracy with the ability to accurately learn new symbol with experience.

References

1. A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *IEEE 11th Int. Conference on Computer Vision*, pages 1–8. IEEE, 2007.
2. 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*, 2(2):58–79, 2013.
3. 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 *Int. Symposium on Experimental Robotics*, June 2014.
4. 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*, 9(Aug):1871–1874, 2008.
5. T.W. Fong, C. Thorpe, and C. Baur. Robot, asker of questions. *Robotics and Autonomous Systems*, 2003.
6. T. Howard, S. Tellex, and N. Roy. A natural language planner interface for mobile manipulators. In *Int. Conference on Robotics and Automation*, June 2014.
7. D.C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.
8. D. Nyga and M. Beetz. Reasoning about unmodelled concepts-incorporating class taxonomies in probabilistic relational models. *arXiv preprint arXiv:1504.05411*, 2015.
9. R. Paul, J. Arkin, N. Roy, and T. Howard. Efficient grounding of abstract spatial concepts for natural language interaction with robot manipulators. In *Proc. of Robotics: Science and Systems (RSS)*, Ann Arbor, Michigan, USA, June 2016.
10. 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 Int. Symposium in Robot and Human Interactive Communication*, pages 570–575, Sept 2010.
11. 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.
12. 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.
13. A. Vedaldi and B. Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. In *Proc. of the 18th ACM Int. Conference on Multimedia*, pages 1469–1472. ACM, 2010.
14. C.L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405. Springer, 2014.