Inferring Maps and Behaviors from Natural Language Instructions

Felix Duvallet*[1], Matthew R. Walter*[2], Thomas Howard*[2], Sachithra Hemachandra*[2], Jean Oh[1], Seth Teller[2], Nicholas Roy[2], and Anthony Stentz[1]

[1] Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, {felixd,jeanoh,tony}@cmu.edu

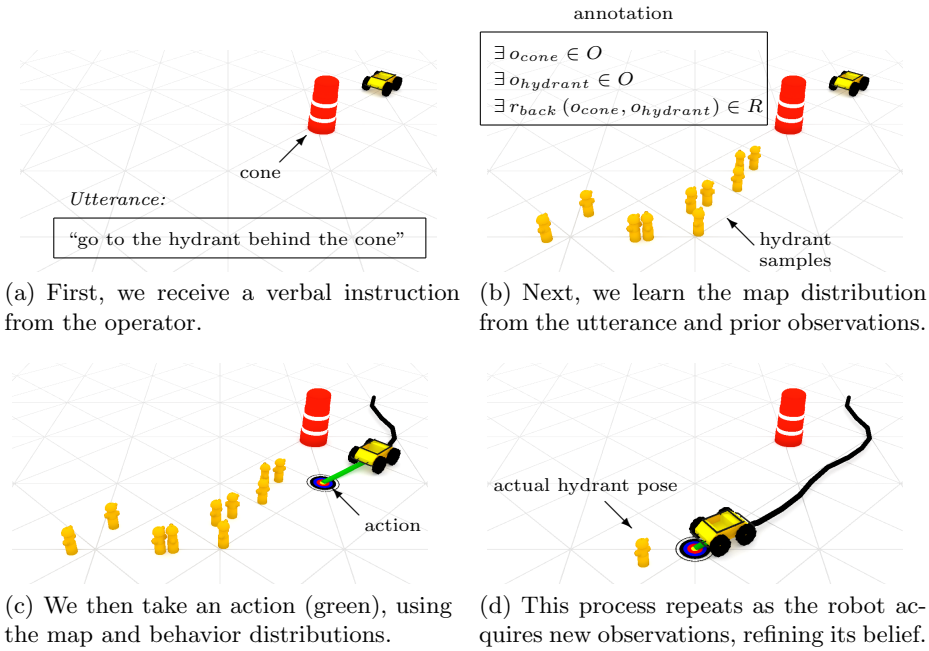
[2] CS & AI Lab, Massachusetts Institute of Technology, Cambridge, MA, USA, {mwalter,tmhoward,sachih,teller,nickroy}@csail.mit.edu

Abstract. Natural language provides a flexible, intuitive way for people to command robots, which is becoming increasingly important as robots transition to working alongside people in our homes and workplaces. To follow instructions in unknown environments, robots will be expected to reason about parts of the environments that were described in the instruction, but that the robot has no direct knowledge about. This paper proposes a probabilistic framework that enables robots to follow commands given in natural language, without any prior knowledge of the environment. The novelty lies in exploiting environment information implicit in the instruction, thereby treating language as a type of sensor which is used to formulate a prior distribution over the unknown parts of the environment. The algorithm then uses this learned distribution to infer a sequence of actions that are most consistent with the command, updating our belief as we gather more metric information. We evaluate our approach through simulation as well as experiments on two mobile robots; our results demonstrate the algorithm's ability to follow navigation commands with performance comparable to that of a fully-known environment.

1 Introduction

Robots are increasingly performing collaborative tasks with people at home, in the workplace, and outdoors, and with this comes a need for efficient communication between human and robot teammates. Natural language offers an effective means for untrained users to control complex robots, without requiring specialized interfaces or extensive user training. Enabling robots to understand natural language instructions would facilitate seamless coordination in human-robot teams. However, interpreting instructions is a challenge, particularly when the robot has little or no prior knowledge of its environment. In such cases, the robot should be capable of reasoning over the parts of the environment that are relevant to understanding the instruction, but may not yet have been observed.

* The first four authors contributed equally to this paper.



(a) First, we receive a verbal instruction from the operator.

(b) Next, we learn the map distribution from the utterance and prior observations.

(c) We then take an action (green), using the map and behavior distributions.

(d) This process repeats as the robot acquires new observations, refining its belief.

Fig. 1. Visualization of one run for the command "go to the hydrant behind the cone," showing the evolution of our beliefs (the possible locations of the hydrant) over time.

Oftentimes, the command itself provides information about the environment that can be used to hypothesize suitable world models, which can then be used to generate the correct robot actions. For example, suppose a first responder instructs a robot to "navigate to the car behind the building," where the car and building are outside the robot's field-of-view and their locations are not known. While the robot has no a priori information about the environment, the instruction conveys the knowledge that there is likely one or more buildings and cars in the environment, with at least one car being "behind" one of the buildings. The robot should be able to reason about the car's possible location, and refine its prior as it carries out the command (e.g., update the car's possible location when it observes a building).

This paper proposes a method that enables robots to interpret and execute natural language commands that refer to unknown regions and objects in the robot's environment. We exploit the information implicit in the user's command to learn an environment model from the natural language instruction, and then solve for the policy that is consistent with the command under this world model. The robot updates its internal representation of the world as it makes new metric observations (such as the location of perceived landmarks) and updates its policy appropriately. By reasoning and planning in the space of beliefs over object locations and groundings, we are able to reason about elements that are not

initially observed, and robustly follow natural language instructions given by a human operator.

More specifically, we describe in our approach (Section 3) a probabilistic framework that first extracts annotations from a natural language instruction, consisting of the objects and regions described in the command and the given relations between them (Fig. 1(a)). We then treat these annotations as noisy sensor observations in a mapping framework, and use them to generate a distribution over a semantic model of the environment which also incorporates observations from the robot's sensor streams (Fig. 1(b)). This prior is used to ground the actions and goals from the command, resulting in a distribution over desired behaviors. This is then used to solve for a policy which yields an action that is most consistent with the command, under the map distribution so far (Fig. 1(c)). As the robot travels and senses new metric information, it updates its map prior and inferred behavior distribution, and continues to plan until it reaches its destination (Fig. 1(d)).

We evaluate our algorithm in Section 4 through a series of simulation-based and physical experiments on two mobile robots that demonstrate its effectiveness at carrying out navigation commands, as well as highlight the conditions under which it fails. Our results indicate that exploiting the environment knowledge implicit in a natural language instruction enables us to predict a world model upon which we can successfully estimate the action sequence most consistent with the command, approaching performance levels of complete environment a priori knowledge. These results suggest that utilizing information implicitly contained in natural language instructions can improve collaboration in human-robot teams.

2 Related Work

Natural language has proven to be effective for commanding robots to follow route directions [1–5] and manipulate objects [6]. The majority of prior approaches require a complete semantically-labeled environment model that captures the geometry, location, type, and label of objects and regions in the environment [2, 5, 6]. Understanding instructions in unknown environments is often more challenging. Previous approaches have either used a parser that maps language directly to plans [1, 3, 4], or trained a policy that reasons about uncertainty and can backtrack when needed [7]. However, none of these approaches directly use the information contained in the instruction to inform their environment representation or reason about its uncertainty. We instead treat language as a sensor that can be used to generate a prior over the possible locations of landmarks by exploiting the information implicitly contained in a given instruction.

State-of-the-art semantic mapping frameworks focus on using the robot's sensor observations to update its representation of the world [8–10]. Some approaches [10] integrate language descriptions to improve the representation but do not extend the maps based on natural language. Our approach treats natural language as another sensor and uses it to extend the spatial representation by

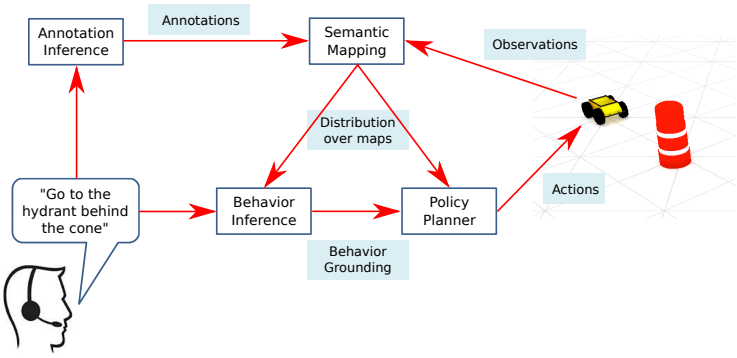


Fig. 2. Framework outline.

adding both topological and metric information, which is then used for planning. Williams et. al. [11] use a cognitive architecture to add unvisited locations to a partial map. However, they only reason about topological relationships to unknown places, do not maintain multiple hypothesis, and make strong assumptions about the environment limiting the applicability to real robot systems. In contrast, our approach reasons both topologically and metrically about objects and regions, and can deal with ambiguity, which allows us to operate in challenging environments.

As we reason in the space of distributions over possible environments, we draw from strategies in the belief-space planning literature. Most importantly, we represent our belief using samples from the distribution, similar to work by Platt et. al. [12]. Instead of solving the complete Partially-Observable Markov Decision Process (POMDP), we instead seek efficient approximate solutions [13, 14].

3 Technical Approach

Our goal is to infer the most likely future robot trajectory $x_{t+1:T}$ up to time horizon T, given the history of natural language utterances Λ^t, sensor observations z^t, and odometry u^t,

$$\arg \max_{x_{t+1:T} \in \Re^n} p\left(x_{t+1:T} | \Lambda^t, z^t, u^t\right). \tag{1}$$

Inferring the maximum a posteriori trajectory (1) for a given natural language utterance is challenging without knowledge of the environment for all but trivial applications. To overcome this challenge, we introduce a latent random variable S_t that represents the world model as a *semantic map* that encodes the location, geometry, and type of the objects within the environment. This allows us to factor the distribution as

$$\arg \max_{x_{t+1:T} \in \Re^n} \int_{S_t} p(x_{t+1:T} | S_t, \Lambda^t, z^t, u^t)\, p(S_t | \Lambda^t, z^t, u^t)\, dS_t. \tag{2}$$

As we maintain the distribution in the form of samples, this simplifies to,

$$\arg \max_{x_{t+1:T} \in \Re^n} \sum_i p(x_{t+1:T} | S_t^{(i)}, \Lambda^t, z^t, u^t)\, p(S_t^{(i)} | \Lambda^t, z^t, u^t) \tag{3}$$

Our algorithm learns these distributions online based upon the robot's sensor and odometry streams and the user's natural language input. We do so through a filtering process whereby we first infer the distribution over the world model S_t based upon annotations identified from the utterance (second term in the integral in (2)), upon which we then infer the constraints on the robot's action that are most consistent with the command given the initial map. At this point, the algorithm solves for the most likely policy under the learned distribution over trajectories (first term in the integral in (2)). During execution, we continuously update the semantic map S_t as sensor data arrives and refine the optimal policy according to the re-grounded language.

We use the Distributed Correspondence Graph (DCG) model [5] to efficiently convert unstructured natural language to symbols that represent the spaces of annotations and behaviors. The DCG model is a probabilistic graphical model composed of random variables that represent language λ, groundings γ, and correspondences between language and groundings ϕ and factors f. Each factor f_{i_j} in the DCG model is influenced by the current phrase λ_i, correspondence variable ϕ_{i_j}, grounding γ_{i_j}, and child phrase groundings $\gamma_{c_{i_j}}$. The parameters in each log-linear model υ are trained from a parallel corpus of labeled examples for annotations and behaviors in the context of a world model Υ. In each, we search for the unknown correspondence variables that maximize the product of factors:

$$\arg \max_{\phi \in \Phi} \prod_i \prod_j f_{i_j} \left(\phi_{i_j}, \gamma_{i_j}, \gamma_{c_{i_j}}, \lambda_i, \Upsilon, \upsilon \right). \tag{4}$$

An illustration of the graphical model used to represent Equation 4 is shown in Figure 3. In Figure 3 the black squares, white circles, and gray circles represent factors, unknown random variables, and known random variables respectively. It is important to note that each phrase can have a different number of vertically aligned factors if the symbols used to ground particular phrases differ. In this paper we use a binary correspondence variable to indicate the expression or rejection of a particular grounding for a phrase. We construct the symbols used to represent each phrase using only the groundings with a true correspondence and take the meaning of a utterance as the symbol inferred at the root of parse tree.

Figure 2 illustrates the architecture of the integrated system that we consider for evaluation. First, the natural language understanding module infers a distribution over annotations conveyed by the utterance. The semantic map learning method then uses this information in conjunction with the prior annotations and sensor measurements to build a probabilistic model of objects and their relationships in the environment. We then formulate a distribution over robot behaviors using the utterance and the semantic map distribution. Next, the planner computes a policy from this distribution over behaviors and maps.

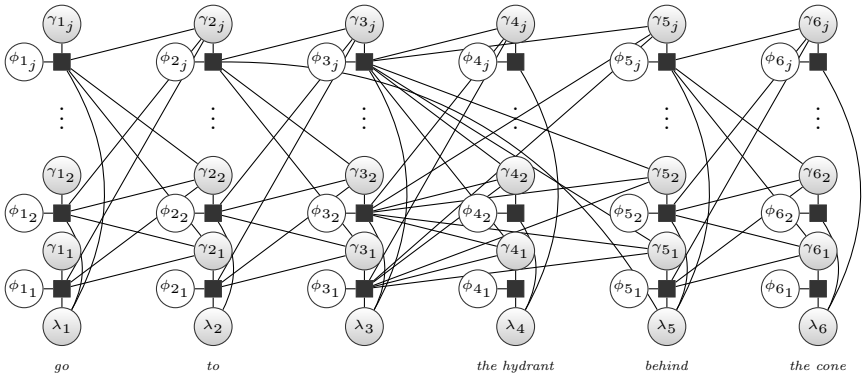


Fig. 3. A DCG used to infer annotations or behaviors from the utterance "go to the hydrant behind the cone." The factors f_{i_j}, groundings γ_{i_j}, and correspondence variables ϕ_{i_j} are functions of the symbols used to represent annotations and behaviors.

As the robot makes more observations or receives additional human input, we repeat the last three steps to continuously update our understanding of the most recent utterance.

3.1 Annotation Inference

The space of symbols used to represent the meaning of phrases in map inference is composed of objects, regions, and relations. Since no world model is assumed when inferring linguistic annotations from the utterance, the space of objects is equal to the number of possible object types that could exist in the scene. Regions are some portion of state-space that is typically associated with a relationship to some object. Relations are a particular type of association between a pair of objects or regions (e.g., front, back, near, far). Since any set of objects, regions, and relations may be inferred as part of the symbol grounding, the size of the space of groundings for map inference grows as the power set of the sum of these symbols. We use the trained DCG model to infer a distribution of annotations α_t from the positively expressed groundings at the root of the parse tree.

3.2 Semantic Mapping

We treat the annotations as noisy observations α_t that specify the existence and spatial relations between labeled objects in the robot's environment. We use these observations along with those from the robot's sensors z_t to learn the distribution over the semantic map $S_t = \{G_t, X_t\}$

$$p(S_t|\Lambda^t, z^t, u^t) \approx p(S_t|\alpha^t, z^t, u^t) \tag{5a}$$

$$= p(G_t, X_t|\alpha^t, z^t, u^t) \tag{5b}$$

$$= p(X_t|G_t, \alpha^t, z^t, u^t)p(G_t|\alpha^t, z^t, u^t), \tag{5c}$$

where the first line follows from the assumption that there is a single annotation α_t for a given utterance λ_t. The last line expresses the factorization into a distribution over the topology and a conditional distribution over the metric map. Owing to the combinatorial number of candidate topologies [10], we employ a sample-based approximation to the latter distribution and model the conditional posterior over poses with a Gaussian, parametrized in the canonical form. In this manner, each particle $S_t^{(i)} = \{G_t^{(i)}, X_t^{(i)}, w_t^{(i)}\}$ consists of a sampled topology $G_t^{(i)}$, a Gaussian distribution over the poses $X_t^{(i)}$, and a weight $w_t^{(i)}$. We note that this model is similar to that of Walter et al. [10], though we don't treat the labels as being uncertain.

We use a Rao-Blackwellized particle filter [15] to efficiently maintain this distribution over time, as the robot receives new annotations and observations while executing the inferred behavior. This process involves proposing updates to each sampled topology that express object observations and annotions. Next, the algorithm uses the proposed topologies to perform a Bayesian update to the Gaussian distribution over the node (object) poses. The algorithm then updates the particle's weight so as to approximate the target distribution. We perform this process for each particle and repeat these steps at each time instance. The following describes each operation in more detail.

During the proposal step, we first augment each sample topology with an additional node and edge that model the robot's motion, resulting in a new topology $S_t^{(i)-}$. We then sample modifications to the graph $\Delta_t^{(i)} = \{\Delta_{\alpha_t}^{(i)}, \Delta_{z_t}^{(i)}\}$ based upon the most recent annotations and sensor observations α_t and z_t

$$p(S_t^{(i)}|S_{t-1}^{(i)}, \alpha_t, z_t, u_t) = p(\Delta_{\alpha_t}^{(i)}|S_t^{(i)-}, \alpha_t)\, p(\Delta_{z_t}^{(i)}|S_t^{(i)-}, z_t)\, p(S_t^{(i)-}|S_{t-1}^{(i)}, u^t),$$

where $S_t^{(i)} = \{S_t^{(i)-}, \Delta_t^{(i)}\}$. The updates can include the addition of nodes to the graph representing newly hypothesized or observed objects. They also may include the addition of edges between nodes to express spatial relations inferred from observations or annotations.

For each language annotation $\alpha_{t,j}$, we sample the graph modifications from the proposal (6) in a multi-stage process.

$$p(\Delta_{\alpha_t}^{(i)}|S_t^{(i)-}, \alpha_t) = \prod_j p(\Delta_{\alpha_{t,j}}^{(i)}|S_t^{(i)-}, \alpha_{t,j}) \tag{6}$$

We use a likelihood model over the spatial relation to sample landmark and figure pairs for the grounding. This model employs a Dirichlet process prior that accounts for the fact that the annotation may refer to existing or new objects. If the landmark and/or the figure are sampled as new objects, we add these objects to the particle, and create an edge between them. We also sample the metric constraint associated with this edge, based on the spatial relation.

When the robot observes objects, a similar process is employed (7). For each observation, a grounding is sampled from the existing model of the world. We add a new constraint to the object when the grounding is valid, and create a

new object and constraint when it is not.

$$p(\Delta_{z_t}^{(i)} | S_t^{(i)-}, z_t) = \prod_j p(\Delta_{z_{t,j}}^{(i)} | S_t^{(i)-}, z_{t,j}) \tag{7}$$

After proposing modifications to each particle, we perform a Bayesian update to their Gaussian distribution. Next, we reweight each particle (8) by taking into account the likelihood of generating language annotations, as well as positive and negative observations of objects. For annotations, we use the natural language grounding likelihood under the map at the previous time step. For object observations, we use the likelihood that the observations were (or were not) generated based upon the previous map. This has the effect of down-weighting particles for which the observations are unexpected.

$$w_t^{(i)} = p(z_t, \alpha_t | S_{t-1})\, w_{t-1}^{(i)} = p(\alpha_t | S_{t-1})\, p(z_t | S_{t-1})\, w_{t-1}^{(i)} \tag{8}$$

We normalize the weights and resample if their entropy exceeds a threshold [15].

3.3 Behavior Inference

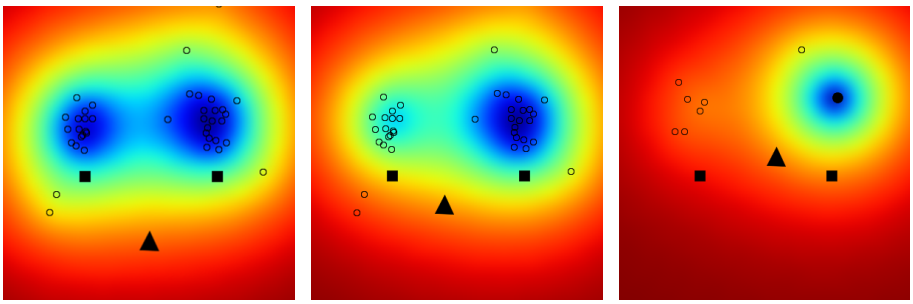
The space of symbols used to represent the meaning of phrases in behavior inference is composed of objects, regions, actions, and goals. Objects and regions are defined in the same manner as in map inference, though the presence of objects is a function of the inferred map. Actions and goals specify how the robot should perform a behavior to the planner. Since any set of actions and goals can be expressed to the planner, the space of groundings for behavior inference also grows as the power set of the sum of these symbols. For the experiments discussed later in Section 4 we assume a number of objects, regions, actions, and goals that are proportional to the number of objects in the hypothesized world model. We use the trained DCG model to infer a distribution of behaviors β from the positively expressed groundings at the root of the parse tree.

3.4 Planner

Since it is difficult to both represent and search the continuum for a trajectory that best reflects the entire instruction in the context of the semantic map, we instead learn a policy that predicts a single action which maximizes the one-step expected value of taking the action a_t from the robot's current pose x_t. This process is repeated until the policy declares it is done following the command using a separate action a_{stop}.

As the robot moves in the environment, it builds and updates a graph of locations it has previously visited, as well as frontiers that lie at the edge of explored space. This graph is used to generate a candidate set of actions that consists of all frontier nodes $\mathcal{F}$ as well as previously-visited nodes $\mathcal{V}$ that the robot can travel to next

$$A_t = \mathcal{F} \cup \mathcal{V} \cup \{a_{\text{stop}}\}. \tag{9}$$



(a) $t = 0$ (b) $t = 4$ (c) $t = 8$

Fig. 4. The value function over time for the command "go to the hydrant behind the cone," where (a) the triangle denotes the robot, squares denote observed cones, and circles denote sampled (empty) and observed (filled) hydrants. The robot first moves towards the left cluster, but after not observing the hydrant, (b) the map distribution peaks at the right cluster. The robot moves right and (c) sees the actual hydrant.

The policy selects the action with the maximum value under our value function

$$\pi(x_t) = \arg \max_{a_t \in A_t} V(x_t, a_t). \tag{10}$$

The value of a particular action is a function of the behavior and the semantic map, which are not observable. Instead, we solve this using the QMDP algorithm [13] by taking the expected value under the semantic map and behavior distributions

$$V(x_t, a_t) \approx \sum_{S_t^{(i)}} \sum_{\beta_j} V\big(x_t, a_t; S_t^{(i)}, \beta_j\big)\, p\big(\beta_j | S_t^{(i)}\big)\, p\big(S_t^{(i)}\big). \tag{11}$$

We define the value for a semantic map particle and behavior as

$$V\big(x_t, a_t; S_t^{(i)}, \beta_j\big) = \gamma^{d(a_t, g_s)}, \tag{12}$$

where γ is the MDP discount factor and d is the Euclidean distance between the action node and the behavior's goal position g_s. Our belief space policy π then picks the maximum value action. We reevaluate this value function as the semantic map and behavior distributions improve with new observations. Figure 4 demonstrates the evolution of the value function over time.

4 Results

We first analyze the ability of our natural language understanding module to independently infer the correct annotations and behaviors for given utterances. Next, we analyze the effectiveness of our end-to-end framework through simulations that consider environments and commands of varying complexity, and

Table 1. Natural language understanding results with 95% confidence intervals.

Model	Accuracy (%)	Training Time (sec)	Inference Time (sec)
Annotation	62.50 (10.83)	145.11 (7.55)	0.44 (0.03)
Behavior	55.77 (6.83)	18.30 (1.02)	0.05 (0.00)

different amounts of prior knowledge. We then demonstrate the utility of our approach in practice using experiments run on two mobile robot platforms. These experiments provide insights into our algorithm's ability to infer the correct behavior in the presence of unknown and ambiguous environments.

4.1 Natural Language Understanding

We evaluate the performance of our natural language understanding component in terms of the accuracy and computational complexity of inference using holdout validation. In each experiment, the corpus was randomly divided into separate training and test sets to evaluate whether the model can recover the correct groundings from the utterance and the world model. Each model used 13,716 features that checked for the presence of words, properties of groundings and correspondence variables, and relationships between current and child groundings and searched the model with a beam width of 4. We conducted 8 experiments for each model type using a corpus of 39 labeled examples of instructions and groundings. For annotation inference we assumed that the space of groundings for every phrase is represented by 8 object types, 54 regions, and 432 relations. For behavior inference we assumed that noun and prepositions ground to hypothesized objects or regions while verbs ground to 2 possible actions, 3 possible modes, goal regions, and constraint regions. In the example illustrated in Fig. 3 with a world model composed of seven hypothesized objects the annotation inference DCG model contained 5,934 random variables and 2,964 factors while the behavior inference DCG model contained 772 random variables and 383 factors. In each experiment 33% of the labeled examples in the corpus were randomly selected for the holdout. The mean number of log-linear model training examples extracted from the 26 randomly selected labeled examples for annotation and behavior inference was 83,547 and 9,224 respectively. Table 1 illustrates the statistics for the annotation and behavior models.

This experiment demonstrates that we are able to learn many of the relationships between phrases, groundings, and correspondences with a limited number of labeled instructions, and infer a distribution of symbols quickly enough for the proposed architecture. As expected the training and inference time for the annotation model is much higher because of the difference in the complexity of symbols. This is acceptable for our framework since the annotation model is only used once to infer a set of observations while the behavior model is used continuously to process the updated map distributions.

4.2 Monte Carlo Simulations

Next, we evaluate the entire framework through an extended set of simulations in order to understand how the performance varies with the environment configuration and the command. We consider four environment templates, with different numbers of figures (hydrants) and landmarks (cones). For each configuration, we sample ten environments, each with different object poses. For these environments, we issued three natural language instructions "go to the hydrant," "go to the hydrant behind the cone," and "go to the hydrant nearest to the cone." We note that these commands were not part of the corpus that we used to train the DCG model. Additionally, we considered six different settings for the robot's field-of-view, 2 m, 3 m, 5 m, 10 m, 15 m, and 20 m, and performed approximately 100 simulations for each combination of environment, command, and field-of-view. As a ground-truth baseline, we performed ten runs of each configuration with a completely known world model.

Table 2. Monte Carlo simulation results with 1σ confidence intervals (Hydrant, Cone).

			Success Rate (%)		Distance (m)	
World	FOV (m)	Relation	Known	Ours	Known	Ours
1H, 1C	3.0	null	100.0	93.9	8.75 (1.69)	16.78 (7.90)
1H, 1C	3.0	"behind"	100.0	98.3	8.75 (1.69)	13.43 (7.02)
1H, 2C	3.0	null	100.0	100.0	11.18 (1.38)	32.54 (18.50)
1H, 2C	3.0	"behind"	100.0	99.5	11.18 (1.38)	40.02 (29.66)
2H, 1C	3.0	null	100.0	54.4	10.49 (1.81)	21.56 (10.32)
2H, 1C	3.0	"behind"	100.0	67.4	10.38 (1.86)	18.72 (10.23)
2H, 1C	5.0	"nearest"	100.0	46.2	9.19 (1.54)	12.05 (5.76)

Table 2 presents the success rate and distance traveled by the robot for these 100 simulation configurations. We considered a run to be successful if the planner stops within 1.5 m of the intended goal. Comparing against commands that do not provide a relation (i.e., "go to the hydrant"), the results demonstrate that our algorithm achieves greater success and yields more efficient paths by taking advantage of relations in the command (i.e., "go to the hydrant behind the cone"). This is apparent in environments consisting of a single figure (hydrant) as well as more ambiguous environments that consist of two figures. Particularly telling is the variation in performance as a result of different fields-of-view. Figure 5 shows how success rate increases and distance traveled decreases as the robot's sensing range increases, quickly approaching the performance of the system when it begins with a completely known map of the environment.

One interesting failure case is when the robot is instructed to "go to the hydrant nearest to the cone" in an environment with two hydrants. In instances where the robot sees a hydrant first, it hypothesizes the location of the cone, and then identifies the observed hydrants and hypothesized cones as being consistent with the command. Since the robot never actually confirms the existence of the

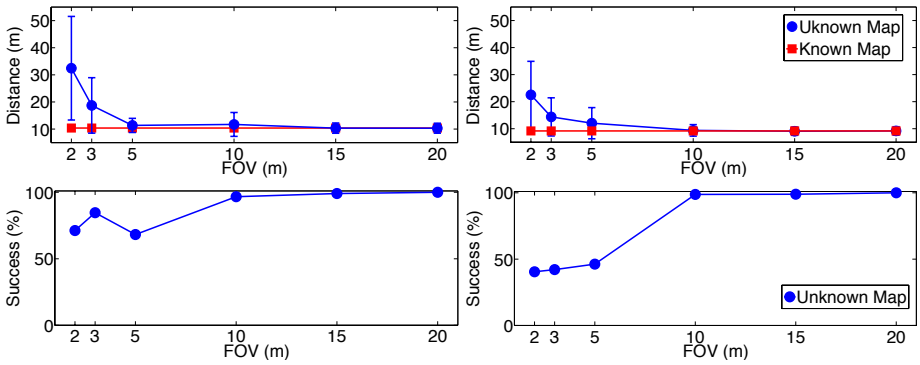


Fig. 5. Distance traveled (top) and success rate (bottom) as a function of the field-of-view for the commands "go to the hydrant behind the cone" (left) and "go to the hydrant nearest to the cone" (right) in simulation.

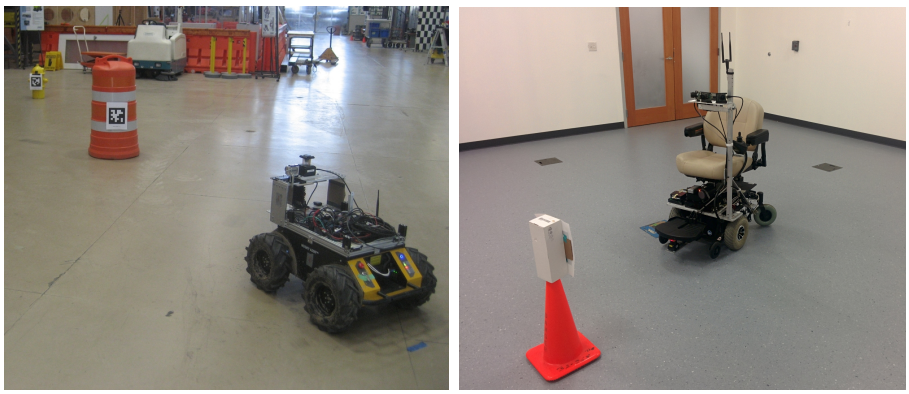
cone in the real world, this results in the incorrect hydrant being labeled as the goal.

4.3 Physical Experiments

We applied our approach to two mobile robots, a Husky A200 mobile robot (Fig. 6(a)) and an autonomous robotic wheelchair [16] (Fig. 6(b)). The use of both platforms demonstrates the application of our algorithm to mobile robots with different vehicle configurations, underlying motion planners, and camera fields-of-view. The actions determined by the planner are translated into lists of waypoints that are handled by each robot's motion planner. We used AprilTag fiducials [17] to detect and estimate the relative pose of objects in the environment, subject to self-imposed angular and range restrictions on the robot's field-of-view.

In each experiment, a human operator issues natural language commands in the form of text that involve (possibly null) spatial relations between one or two objects. The results that follow involve the commands "go to the hydrant," "go to the hydrant behind the cone," and "go to the hydrant nearest to the cone." As with the simulation-based experiments, these instructions did not match those from our training set. For each of these commands, we consider different environments by varying the number and position of the cones and hydrants and by changing the robot's field-of-view. For each configuration of the environment, command, and field-of-view, we perform ten trials with our algorithm. For a ground-truth baseline, we perform an additional run with a completely known world model. We consider a run to be a success when the robot's final destination is within 1.5 m of the intended goal.

Table 3 presents the success rate and distance traveled by the wheelchair for these experiments. Compared to the scenario in which the command does not provide a relation (i.e., "go to the hydrant"), we find that our algorithm is able



(a) Husky (b) Wheelchair

Fig. 6. The setup for the experiments with the (a) Husky and (b) wheelchair platforms.

Table 3. Experimental results with 1σ confidence intervals (Hydrant, Cone).

			Success Rate (%)		Distance (m)	
World	FOV (m)	Relation	Known	Ours	Known	Ours
1H, 1C	2.5	null	100.0	100.0	4.69	16.56 (7.20)
1H, 1C	2.5	"behind"	100.0	100.0	4.69	9.91 (3.41)
1H, 2C	3.0	"behind"	100.0	100.0	4.58	7.64 (2.08)
2H, 1C	2.5	"behind"	100.0	80.0	5.29	6.00 (1.38)
2H, 1C	4.0	"nearest"	100.0	100.0	4.09	4.95 (0.39)
2H, 1C	3.0	"nearest"	100.0	50.0	6.30	7.05 (0.58)

to take advantage of available relations ("go to the hydrant behind the cone") to yield behaviors closer to that of ground truth. The results are similar for the Husky platform, which resulted in an 83.3% success rate when commanded to "go to the hydrant behind the cone" in an environment with one cone and one hydrant. The ability to utilize relation annotations is also important when the same command is given in an environment with two figures (hydrants). Of the ten runs, the robot successfully identified the correct hydrant as the goal eight times, and chose the wrong hydrant for the remaining two. These failures occur when the field-of-view is such that the robot only observes the incorrect hydrant. The semantic map distribution then hypothesizes the existence of cones in front of the hydrant, which leads to a behavior distribution peaked around this goal. In the eight successful trials, the robot observes all three objects and infers the correct behavior. Similarly, if we consider the command "go to the hydrant nearest to the cone," we find that the robot reaches the goal in all ten experiments with a 4 m field-of-view. However, reducing the field-of-view to 3 m results in the robot reaching the goal in only half of the trials.

5 Conclusions

Enabling robots to reason about parts of the environment that have not yet been visited solely from a natural language description serves as one step towards effective and natural collaboration in human-robot teams. By using language as a sensor, we are able to paint a rough picture of what the unvisited parts of the environment could look like, which we utilize during planning and update with actual sensor information during task execution.

Our approach exploits the information implicitly contained in the language to infer the relationship between objects that may not be initially observable, without having to consider those annotations as a separate utterance. By learning a distribution over the map, we generate a useful prior that enables the robot to sample possible hypotheses, representing different environment possibilities that are consistent with both the language and the available sensor data. Learning a policy which reasons in the belief space of these samples achieves a level of performance that approaches full knowledge of the world ahead of time.

These evaluations provide a preliminary validation of our framework. Future work will test the algorithm's ability to handle utterances that present complex relations (e.g., "go to the cone near the tree by the wall") and behaviors that are more detailed (e.g., "go to the cone near the barrel and stay to the right of the car") than those considered above. An additional direction for following work is to explicitly reason over exploratory behaviors that take information gathering actions to resolve uncertainty in the map. Currently, any exploration on the part of the algorithm is opportunistic, which might not be sufficient in more challenging scenarios. Furthermore, for utterances that contain ambiguous information or are difficult to parse, we may be able to use a dialogue system to resolve the ambiguity. For example, the utterance "go to the cone" can be ambiguous when there are several cones present, but "the one nearest to the tree" may provide the missing piece of information to follow the direction correctly.

Acknowledgments

The authors would like to thank Bob Dean for his help with the Husky platform. This work was supported in part by the Robotics Consortium of the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement W911NF-10-2-0016.

Bibliography

[1] MacMahon, M., Stankiewicz, B., Kuipers, B.: Walk the talk: Connecting language, knowledge, and action in route instructions. In: Proc. Nat'l Conf. on Artificial Intelligence (AAAI). (2006)

[2] Kollar, T., Tellex, S., Roy, D., Roy, N.: Toward understanding natural language directions. In: Proc. Int'l. Conf. on Human-Robot Interaction. (2010)

[3] Chen, D.L., Mooney, R.J.: Learning to interpret natural language navigation instructions from observations. In: Proc. Nat'l Conf. on Artificial Intelligence (AAAI). (2011)
[4] Matuszek, C., Herbst, E., Zettlemoyer, L., Fox, D.: Learning to parse natural language commands to a robot control system. In: Proc. Int'l. Symp. on Experimental Robotics (ISER). (2012)
[5] Howard, T., Tellex, S., Roy, N.: A natural language planner interface for mobile manipulators. In: Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA). (2014)
[6] Tellex, S., Kollar, T., Dickerson, S., Walter, M.R., Banerjee, A.G., Teller, S., Roy, N.: Understanding natural language commands for robotic navigation and mobile manipulation. In: Proc. Nat'l Conf. on Artificial Intelligence (AAAI). (2011)
[7] Duvallet, F., Kollar, T., Stentz, A.: Imitation learning for natural language direction following through unknown environments. In: Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA). (2013)
[8] Zender, H., Martínez Mozos, O., Jensfelt, P., Kruijff, G., Burgard, W.: Conceptual spatial representations for indoor mobile robots. Robotics and Autonomous Systems (2008)
[9] Pronobis, A., Martínez Mozos, O., Caputo, B., Jensfelt, P.: Multi-modal semantic place classification. Int'l J. of Robotics Research (2010)
[10] Walter, M.R., Hemachandra, S., Homberg, B., Tellex, S., Teller, S.: Learning semantic maps from natural language descriptions. In: Proc. Robotics: Science and Systems (RSS). (2013)
[11] Williams, T., Cantrell, R., Briggs, G., Schermerhorn, P., Scheutz, M.: Grounding natural language references to unvisited and hypothetical locations. In: Proc. Nat'l Conf. on Artificial Intelligence (AAAI). (2013)
[12] Platt, R., Kaelbling, L., Lozano-Perez, T., Tedrake, R.: Simultaneous localization and grasping as a belief space control problem. In: Proc. Int'l. Symp. of Robotics Research (ISRR). (2011)
[13] Littman, M.L., Cassandra, A.R., Kaelbling, L.P.: Learning policies for partially observable environments: Scaling up. In: Proc. Int'l Conf. on Machine Learning (ICML). (1995)
[14] Roy, N., Burgard, W., Fox, D., Thrun, S.: Coastal navigation-mobile robot navigation with uncertainty in dynamic environments. In: Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA). (1999)
[15] Doucet, A., de Freitas, N., Murphy, K., Russell, S.: Rao-Blackwellised particle filtering for dynamic Bayesian networks. In: Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI). (2000)
[16] Hemachandra, S., Kollar, T., Roy, N., Teller, S.: Following and interpreting narrated guided tours. In: Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA). (2011)
[17] Olson, E.: AprilTag: A robust and flexible visual fiducial system. In: Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA). (May 2011)