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 speech and text provides a flexible, intuitive means to issue commands to robots. A robot's ability to interpret these natural language directives is becoming critical as robots work alongside people in our homes, hospitals, and workplaces. One class of solutions frame the language understanding problem as one of inferring the robot actions and objects and locations in the environment that are associated with a given free-form instruction. These approaches, however, require that this world model representation of the environment is fully known. This paper proposes a probabilistic framework that enables robots to successfully follow natural language commands without any prior knowledge of its operating environment. The novelty lies in exploiting environment knowledge implicit in the instruction to predict a world model upon which we can estimate the states and actions most consistent with the command. Specifically, the algorithm learns a distribution over the metric and semantic properties of the environment based upon annotations inferred from the command. It uses this distribution to then infer a distribution over the corresponding behaviors and executes a policy that yields the most likely action under this distribution. We demonstrate the algorithms ability to follow natural language navigation commands within a priori unknown environments.

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. Owing to its flexibility and intuitiveness, natural language has proven to be an effective means for people to command and control robots, as demonstrated by recent work that allows robots to follow a user's spoken instructions that command navigation [1–8] and object manipulation [8, 9].

A common approach to natural language understanding for robots is to perform what Harnad [10] refers to as the symbol grounding problem in which

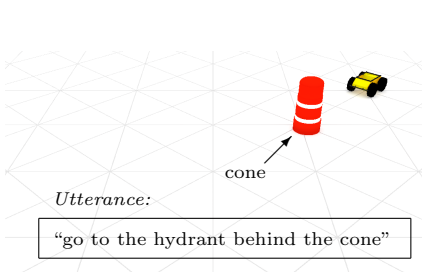
* The first four authors contributed equally to this paper

linguistic elements from free-form commands are mapped to their corresponding manifestation in the external world. Most existing solutions assume to have access to an a priori known *world model* that expresses the space of objects, locations, and actions available to the robot that may serve as referents for the command. The problem of natural language understanding is then one of inferring the elements from this world model that are most likely associated with a given utterance [5, 9]. However, there are many scenarios in which the robot will have limited to no prior knowledge of the environment, and the task of requiring a user to manually provide a sufficiently complete world model may be overly burdensome or impossible. Oftentimes, the command itself provides information about the environment that can be used to hypothesize suitable world model upon which the relevant actions of the robot can be grounded. For example, suppose that a user instructs a robot to "Navigate to the car behind the building," where the car and location are outside the robot's field-of-view and their location unknown. This 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 hypothesis as it carries out the command (e.g., when it observes a building).

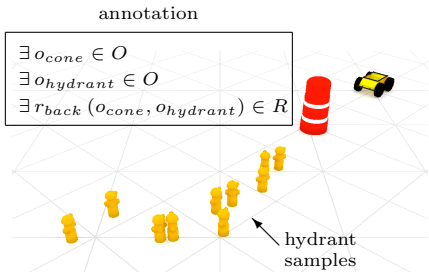
This paper proposes a method that enables robots to interpret and execute natural language commands that refer to unknown regions and objects in the robot's environment. We address the problem by exploiting the existence of annotations implicit in the user's command to simultaneously learn an environment model from the natural language, 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 gathers metric information, such as the location of perceived landmarks. Specifically, we propose a probabilistic framework that infers a distribution over a semantic model of the environment and the grounded behavior given a natural-language command and observations captured by the robot's sensor streams. The framework then uses this distribution over the world model and behaviors to solve for a policy that yields actions with the highest likelihood of being consistent with the command. By reasoning and planning in the space of beliefs over landmark and object locations, we are able to reason about elements that are not initially observed, and robustly follow natural language instructions given by a human operator. We evaluate our algorithm through a series of simulation-based and physical experiments that demonstrate its effectiveness at carrying out navigation commands, as well as the conditions under which it fails.

2 Related Work

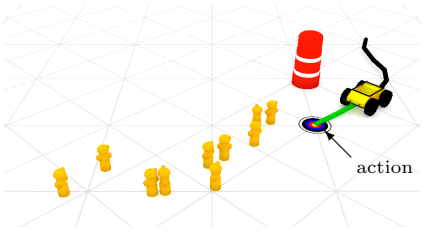
Commanding robots using natural language has proven to be effective for robots performing simple tasks such as following route directions [1–3, 5–8] and manipulating objects [8, 9]. With the exception of the work by [2] and [7], existing approaches require an a priori known environment model that captures the ge-



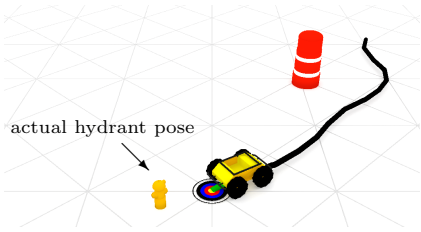
(a) First, we receive a verbal instruction from the operator.



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



(c) We then take one action (green line), using the map distribution and the most likely cost functions from the command.



(d) This process is repeated, and the robot incorporates new observations. As the map distribution is refined, the robot moves to wards the goal.

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. For clarity, we have left out the covariance ellipses of the hypothesized landmarks.

ometry, location, type, and label of objects and regions within the environment. While our approach is able to incorporate a prior distribution over the world model, it is specifically designed to function with no previous knowledge of the environment. Instead, we infer annotations present in the free-form command and exploit these annotations to learn a distribution over the world model that we then opportunistically refine as the robot carries out the resulting behavior.

Some approaches are able to follow natural language directions through unknown environments by using a parser to map language to actions [6, 7, 2]. However, these approaches do not reason directly about the environment, and as such cannot reason about mistakes if the environment does not match the command. We instead leverage the available information in the command to generate a prior over the possible locations of landmarks, act within this distribution, and refine our estimate as we gain more information about the environment.

Duvallet et. al. [11] trains a policy to follow directions through unknown environments, reasoning about uncertainty and backtracking when the policy makes a mistake. However, the information contained in the utterance about the unobserved parts of the environments is not used directly, and they do not reason

about landmarks that have not yet been detected. Our work treats language as a sensor that can be used to generate maps.

Williams et. al. [12] use a cognitive architecture to add unvisited locations to a map. However, they only reason about topological relationships between unknown places, only operate indoors, and do not reason about multiple landmarks of the same type. Our approach reasons both topologically and metrically, and can deal with ambiguous environments.

Our work also draws from related work in exploration strategies for Simultaneous Localization and Mapping (SLAM), where we must gather information to improve our model of the environment [13]. However, our goal is to follow the direction correctly (not reduce the uncertainty in the map or robot pose), and thus we explore only as much as necessary to complete the task.

As we are reasoning 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. [14]. Instead of solving the complete Partially-Observable Markov Decision Process (POMDP), we instead seek efficient approximate solutions [15, 16].

3 Technical Approach

Our goal is to infer the most likely robot trajectory $\mathbf{x}(t)$ given the history of natural language utterances Λ^t, sensor observations z^t, and odometry u^t,

$$\underset{\mathbf{x}(t) \in \Re^n}{\arg\max}\, p\left(\mathbf{x}(t) \,|\Lambda^t, z^t, u^t\right). \tag{1}$$

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

$$\underset{\mathbf{x}(t) \in \Re^n}{\arg\max} \int_{S_t} p(\mathbf{x}(t)|S_t, \Lambda^t, z^t, u^t)\, p(S_t|\Lambda^t, z^t, u^t)\, dS_t. \tag{2}$$

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

$$\underset{\mathbf{x}(t) \in \Re^n}{\arg\max} \sum_i p(\mathbf{x}(t)|S_t^{(i)}, \Lambda^t, z^t, u^t)\, p(S_t^{(i)}|\Lambda^t, z^t, u^t) \tag{3}$$

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

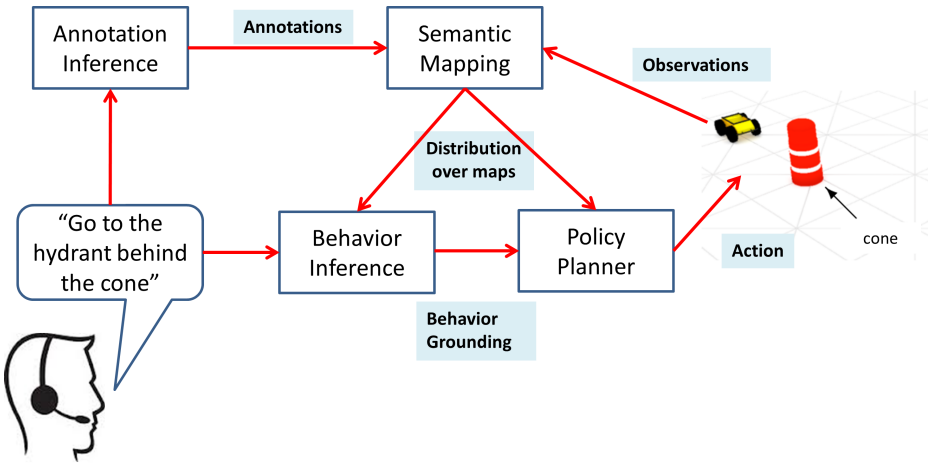


Fig. 2. Framework outline.

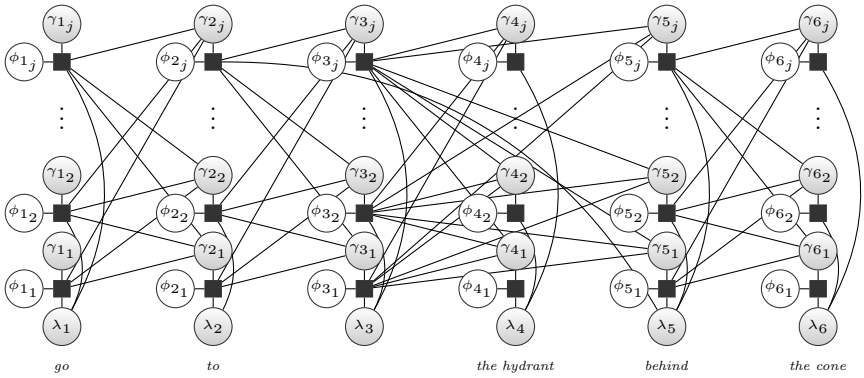
that are most consistent with the command given the initial map. At this point, the algorithm solves for the most likely policy under the learned distribution over trajectories (first term in the integral in (2)). During execution, we continuously update the semantic map S_t as sensor data arrives and refine the optimal policy according to the re-grounded language.

We use the Distributed Correspondence Graph (DCG) model [8] 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 that are represented by log-linear models. The parameters in each log-linear model is 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:

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

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.

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 graph then uses this information in conjunction with the prior utterances and sensor measurements to build a probabilistic model of objects and their relationships



(a) DCG model for "go to the hydrant behind the cone"

Fig. 3. An illustration of two individual DCG models used to infer annotations and behaviors from the instruction "go to the hydrant behind the cone". Each model is trained from separate parallel corpora of examples and is constructed from different sets of symbol groundings.

in the environment. We then formulate a distribution over robot behaviors using the utterance and the semantic graph distribution. Next, the planner computes a policy from this distribution over behaviors and maps. 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 interpreting the utterance for linguistic observations, 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. For the experiments discussed later in Section 4 we assume that the space of groundings for every phrase is represented by 8 objects, 54 regions, and 432 relations. We use a DCG model trained from a parallel corpus of utterances and annotations to infer a distribution of hypothetical observations that the semantic mapping process will fuse with information from other sensors.

3.2 Semantic Mapping

We treat the annotations as noisy observations z_t^a that specify the existence and relative pose of labeled objects in the robot's environment. We use these observations along with those from the robot's onboard sensors z_t^o to learn the distribution over the semantic map, $S_t = \{G_t, X_t\}$

$$p(S_t|\Lambda^t, \{z^o\}^t, u^t) \approx p(S_t|\{z^a\}^t, \{z^o\}^t, u^t) \tag{5a}$$
$$= p(G_t, X_t|\{z^a\}^t, \{z^o\}^t, u^t) \tag{5b}$$
$$= p(X_t|G_t, \{z^a\}^t, \{z^o\}^t, u^t)p(G_t|\{z^a\}^t, \{z^o\}^t, u^t), \tag{5c}$$

where the first line follows from the assumption that there is a single annotation z_t^a 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 [17], we employ a sample-based approximation to this 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. [17], though we don't treat the labels as being uncertain.

We use a Rao-Blackwellized particle filter [18] to efficiently maintain this distribution over time, as the robot receives new observations while executing the inferred behavior. At a high level, this process involves proposing updates to each sampled topology that express observed objects as well as spatial relations between hypothesized objects based language-based annotations. 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 $S_t^{(i)}$ 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 $\Delta S_t^{(i)}$ based upon the most recent annotations and sensor observations z_t^a and z_t^o.

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

The updates can include the addition of new nodes to the graph that represent newly hypothesized or observed objects. They also may include the addition of edges to existing nodes that express spatial relations based on the robot's observation of an object already in the map.

For each language annotation $z_{t(j)}^a$, the graph modifications are sampled from the proposal (6) in a multi-stage process.

$$p(\Delta S_t^{(i)}|S_t^{(i)-}, z_t^a) = \prod_j p(\Delta S_t^{(i)}|S_t^{(i)-}, z_{t(j)}^a) \tag{6}$$

Firstly, a grounding is sampled from existing valid landmark and figure pairs based on the likelihood of the spatial relation (for the given pair) using a Dirichlet process prior (where the count is the likelihood of the spatial relation). If this results in an ungrounded relation, the second stage of sampling samples a landmark and figure object using a Dirichlet process prior (with each object of the same type having a count of 1). If the landmark and/or the figure are sampled as new objects, we create these objects in the world model and create an edge between the two objects. We also sample the constraint for this edge based on the spatial relation (using a likelihood function for the spatial relation and rejection sampling).

When the robot observes objects using its sensors, a similar process is employed (7). For each observation, a grounding is sampled from the existing model of the world. If this results in a valid grounding, we add a new constraint to this object, while if this results in a new object, we create this object in the map and add the constraint.

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

After each particle has been modified, we also update its weight. The update (8) takes in to account the likelihood of generating language annotations, as well as positive and negative observations of objects, given each particle at the previous timestep. For language groundings, this is the grounding likelihood for natural language utterances before the map was updated. For object observations, this is the likelihood of the given map generating observations (or not) given the current position of the robot and the robot's field-of-view. This would down-weight particles that have objects within the field-of-view when the robot did not observe an object as well as when the robot observed an object (but the particle did not have an object near that location).

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

Once the weights are normalized, particles are resampled if needed.

3.3 Behavior Inference

The space of symbols used to represent the meaning of phrases in behavior inference is composed of objects, regions, actions, modes, constraints, and goals. Objects and regions are defined in the same manner as map inference though the presence of objects is a function of the inferred map. Actions, modes, constraints, and goals are a specification to a planner that dictates how the robot should perform a behavior. Since any set of actions, modes, constraints, 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 3 types of actions, 3 types of modes, and a number of objects, regions, goals, and constraints that are proportional to the number of objects in the hypothesized environment. We use a DCG model trained form a

parallel corpus of utterances and behaviors to infer a distribution of behaviors for each particle that we receive from the semantic graph and pass this information to the planner.

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 Graph, we instead learn a policy that predicts a single action which maximizes the one-step expected value of taking the action a from the robot's current pose $\mathbf{x}(t)$. This process is repeated until the policy declares it is done following the direction using a separate action a_{stop}.

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

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

Each action represents a node in the planner's topology that the robot can travel to next.

Each action is evaluated under the policy, which maximizes the value of that action under our current distribution of maps:

$$\pi\left(\mathbf{x}(t), \Phi\right) = \underset{a \in A_x}{\operatorname{argmax}}\ V\left(S, a, \Phi\right). \tag{10}$$

Solving the complete POMDP would scale poorly with the number of hypothesizes, which would be incongruent with a fast replanning cycle. We instead use the QMDP algorithm, which is an efficient approximate POMDP algorithm and approximates the POMDP belief value function by assuming (falsely) that the belief state will become fully observable after one step [19]. This enables us to approximate the value function as an expectation over MDP value functions as

$$V(S, a) \approx \sum_{s \in S} p(s) V(s, a). \tag{11}$$

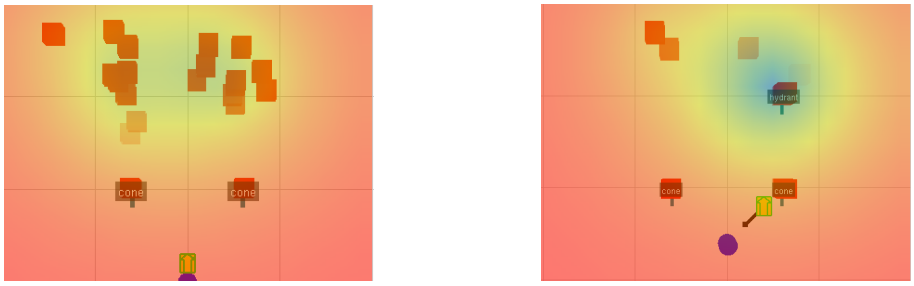
The value of a single sampled world $V\left(s, a, \Phi\right)$ is inferred from the natural language expression and the Semantic Graph

$$V(s, a) = \gamma^{d(a, g_s)}, \tag{12}$$

where γ is the MDP discount factor and d is the Euclidean distance between the action endpoint (where the robot would be after completing the action) and the goal position (given by the annotations Φ).

Our belief space policy π seeks to pick the action which maximizes the value from the current position $\mathbf{x}(t)$

$$\pi\left(\mathbf{x}(t), \Phi\right) = \underset{a \in A_x}{\operatorname{argmax}} \sum_{s \in S} p(s) V(s, a). \tag{13}$$



(a) (b)

Fig. 4. A visualization of the value function evolving over time. (a) The cones are initially visible and the robot has inferred the location of the hydrant in two regions. The robot first travels to the left-most region and when the map is updated to reflect the absence of the hydrant, the value function peaks around the region to the right. (c) The robot observes the cone, which concentrates the value.

While this approach is inherently greedy, we believe it has several advantages that make it well suited for our approach. Since it is fast to replan in the presence of new information, we can use of the fact that the world likelihoods $p(s)$ will change and that particles will be resampled in new locations as the robot moves and observes new areas. This will change the value function, which will in turn alter the behavior of the robot. This is shown in Figure 4, where the value function evolves as the robot moves in an environment containing two cones and one hydrant. Using a more sophisticated POMDP solver is part of our future work, and would enable us to take purely information-gathering actions.

4 Results

We demonstrate the utility of our approach in practice using experiments run on two different mobile robot platforms. We also evaluate the effect of *a priori* knowledge and different environments on our approach by performing a large number of simulation experiments. These simulations provide a better statistical analysis of the algorithm for various environment conditions.

4.1 Natural Language Understanding

To evaluate the ability of the natural language understanding component of our framework we measured the accuracy and computational complexity of probabilistic inference using holdout validation. In each experiment the corpus was randomly divided into a set of examples that was used to train a model and a set of examples that were used to evaluate if the model could recover the correct groundings from only 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

(a) Operator viewpoint.

(b) Robot viewpoint.

Fig. 5. Difference in viewpoints between the robot and operator. To understand the utterance "go to the hydrant behind the cone", the robot must reason about the possible locations for the hydrant even though it cannot be detected.

groundings. We conducted 8 experiments for each type of model using a corpus of 36 labeled examples of instructions and groundings. Statistics for the average accuracy, training time, and inference time for the annotation and behavior models are illustrated in Table 1.

Table 1. Statistics for average accuracy, training time, and inference time for the annotation and behavior models from the natural language understanding experiments.

Model	Accuracy (%)	Training Time (sec)	Inference Time (sec)
Annotation	60.42	127.411	0.42
Behavior	57.29	16.062	0.05

The training time and inference time for the annotation model is as expected much more significant because the vastly larger number of groundings for phrases in the model. 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 continuously processing updated map distributions.

4.2 Physical Experiments

We applied our approach two mobile robots, a Husky A200 mobile robot and an autonomous robotic wheelchair [20]. The use of both platforms shows the robustness of our approach to different vehicle configurations, underlying motion planners, and camera fields of view. The robots are commanded by sending a list of waypoints to their respective motion planners. Perception of landmarks is done using the AprilTag fiducial system for object detection and localization [21], but

we have the ability to integrate a semantic perception system that uses camera images and 3D point clouds in a future version of the system.

In the experiments, a human operator instructs the robot to "drive to the hydrant behind the cone." The object of the sentence is clearly visible from the viewpoint of the operator, however the hydrant is out of the sensing range, and occluded by the orange cone. To vary the difficulty of the task, we ran the same command on both a simple environment and a complex one. In the simple environment, only one landmark of each type was present. Figure 5 shows an example of the simple environment. In the complex environment, two cones were present, but only one hydrant.

We measure the success rate of each trial (whether or not the robot ends within N meters of the goal), as well as the distance traveled by the robot to reach its destination. As abaseline for the experiments, we compare the performance of the system to one that operates with full knowledge of the environment. This knowledge is acquired by manually driving the robot around the test environment while the semantic graph builds a map from camera observations. This provides the robot with full knowledge of the location of all landmarks.

These mobile robot experiments provide insights into the ability of our approach to infer missing information by using the utterance, reason about the uncertainty in the landmark locations, and handle ambiguous landmarks in the case where the environment contains several landmarks of the same type.

Simple Environment We performed 12 experiments in which we gave the robot natural language instructions that involved a spatial relation between a pair of objects (e.g., "go to the hydrant behind the cone"). Of these commands, half were given with the map known a priori and the other half with an unknown map. The language models used to infer the observations and behaviors were trained from a more general corpus of 22 different annotated utterances. The results, illustrated below in Table 2, show that our approach executes the desired behavior at a rate of 83.3% compared to the full-knowledge case, even though the object that determines the destination is not initially visible nor is it known in advance. As we expected, the time required for execution was greater in the case that the map was unknown due to the exploratory behavior that is performed. The robot executed the intended action at a cost that is about 28% more than that of the fully known map. Illustrations of one run in an unknown map with the utterance "go to the hydrant behind the cone" at various periods of the test are shown in Figure 1.

Table 2. Mean time and distance with 95% confidence intervals over 12 runs (6 per experimental condition) and success rates for the different scenarios.

Algorithm	Time (sec)	Distance (m)	Success (%)
Known Map	26.4 ± 4.2	8.4 ± 1.3	100.0
Unknown Map	34.0 ± 18.6	8.1 ± 0.6	83.3

Table 3. Wheelchair Experimental Results

Scenario	Algorithm	# Runs	Success (%)	Distance (m)
1	Known Map	5	100.00	8.01 ± 1.09
1	Unknown Map	100	100.00	14.95 ± 9.89
1	Coverage	?	?	? ± ?
2	Known Map	5	100.00	9.17 ± 0.15
2	Unknown Map	89	81.82	34.57 ± 37.90
2	Coverage	?	?	? ± ?

Table 4. Monte Carlo Simulation Results

Scenario	Algorithm	# Runs	Success (%)	Distance (m)
1	Known Map	5	100.00	8.01 ± 1.09
1	Unknown Map	100	100.00	14.95 ± 9.89
1	Coverage	?	?	? ± ?
2	Known Map	5	100.00	9.17 ± 0.15
2	Unknown Map	89	81.82	34.57 ± 37.90
2	Coverage	?	?	? ± ?

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. We expect this to be necessary in more challenging scenarios. Furthermore, or 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" may not contain enough information 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] Skubic, M., Perzanowski, D., Blisard, S., Schultz, A., Adams, W., Bugajska, M., Brock, D.: Spatial language for human-robot dialogs. IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews **34**(2) (2004) 154–167

[2] 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) 1475–1482

[3] Dzifcak, J., Scheutz, M., Baral, C., Schermerhorn, P.: What to do and how to do it: Translating natural language directives into temporal and dynamic logic representation for goal management and action execution. In: Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA). (2009) 4163–4168

[4] Matuszek, C., Fox, D., Koscher, K.: Following directions using statistical machine translation. In: Proc. Int'l. Conf. on Human-Robot Interaction. (2010)

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

[6] 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) 859–865

[7] 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)

[8] 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)

[9] 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) 1507–1514
[10] Harnad, S.: The symbol grounding problem. Physica D **42** (1990) 335–346
[11] 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), Karlsruhe, Germany (May 2013) 1047–1053
[12] 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), Bellevue, WA (2013) 947–953
[13] Stachniss, C., Grisetti, G., Burgard, W.: Information gain-based exploration using rao-blackwellized particle filters. In: Proc. Robotics: Science and Systems (RSS), Cambridge, MA (June 2005)
[14] 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), Flagstaff, AZ (August 2011)
[15] 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), Tahoe City, CA (July 1995)
[16] 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), Detroit, MI (May 1999) 35–40
[17] 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), Berlin, Germany (June 2013)
[18] 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) 176–183
[19] Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press (2005)
[20] 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) 2574–2579
[21] Olson, E.: AprilTag: A robust and flexible visual fiducial system. In: Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA). (May 2011)