Learning Qualitative Spatial Relations for Robotic Navigation

Abdeslam Boularias
Department of Computer Science
Rutgers University

Felix Duvallet
École Polytechnique
Fédérale de Lausanne (EPFL)

Jean Oh and **Anthony Stentz**[*]
Robotics Institute
Carnegie Mellon University

Abstract

We consider the problem of robots following natural language commands through previously unknown outdoor environments. A robot receives commands in natural language, such as "Navigate around the building to the car left of the fire hydrant and near the tree." The robot needs first to classify its surrounding objects into categories, using images obtained from its sensors. The result of this classification is a map of the environment, where each object is given a list of semantic labels, such as "tree" or "car", with varying degrees of confidence. Then, the robot needs to *ground* the nouns in the command, *i.e.*, mapping each noun in the command into a physical object in the environment. The robot needs also to ground a specified navigation mode, such as "navigate quickly" or "navigate covertly", as a cost map. In this work, we show how to ground nouns and navigation modes by learning from examples demonstrated by humans.

1 Introduction

We consider the problem of commanding mobile robots in unknown, semi-structured, outdoor environments using natural language. This problem arises in human-robot teams, where natural language is a favored communication means. Therefore, robots need to understand the environment from the standpoint of their human teammates, and to translate instructions received in natural language into plans.

For example, to execute the command "Navigate around the building to the car that is left of the fire hydrant and near the tree", the robot needs to find out which objects in the environment are meant by "building" and "car", and to plan a path accordingly. To accomplish this goal, the robot needs to *ground* all the nouns in the command ("building", "car", "fire hydrant" and "tree") into specific objects in the environment, and to interpret the spatial relations ("left of" and "near") and the navigation mode ("around").

The robot must first recognize the objects in the environment and to label them. We use the semantic perception

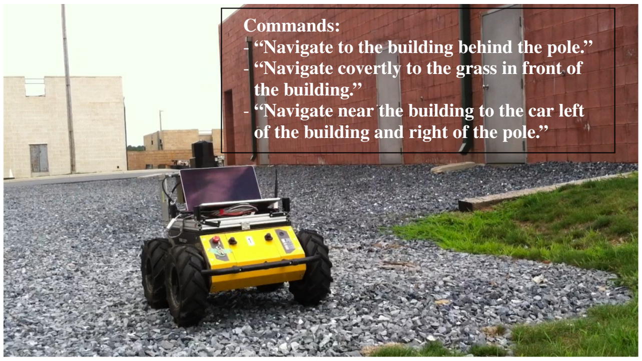

Commands:
- "Navigate to the building behind the pole."
- "Navigate covertly to the grass in front of the building."
- "Navigate near the building to the car left of the building and right of the pole."

Figure 1: Clearpath$^{\text{TM}}$ Husky robot used in the experiments

method proposed in [Munoz, 2013] which has been proven effective in outdoor environments [Munoz *et al.*, 2010]. The semantic perception module receives scene images from a 2D camera and classifies each object into categories with different confidence values.

Grounding is performed by combining the labels obtained from semantic perception with the spatial relations obtained from parsing the command. There are three types of uncertainty that make grounding a challenging task. First, objects are often misclassified because of occlusions and noise in the sensory input. Classification errors also occur when the environment contains objects that are significantly different from the ones used for training the classifier. Second, the commands can be ambiguous, *i.e.*, multiple objects may satisfy the constraints in a given command. Third, spatial relations are often subjectively interpreted. People have different views on what "left of a building" is, for example.

To trade off these uncertainties, we use a Bayesian model for grounding. Our approach is based on using the confidence values of the perception as a prior distribution on the true category of each object. A posterior joint distribution on the objects is computed based on how well each object satisfies the spatial constraints. A key component of this model is a function that maps two objects and a spatial relation into a probability. This function is learned from annotated examples. We also learn a function that maps a navigation mode into a cost map for path planning, using Imitation Learning [Ratliff *et al.*, 2006]. Learning by imitation enables the robot to in-

[*] This work was done while all the authors were with the Robotics Institute of Carnegie Mellon University.

terpret commands according to the subjective definitions of its human user. Moreover, navigation modes as "navigate covertly" do not have clear definitions that can be used for handcrafting a path cost function.

Finally, we compute a joint distribution on goal objects and on landmark objects used for specifying a navigation mode, so that results with small path costs have high probabilities. For path planning, we use the cost map based planner, PMAP with Field D* [Stentz, 1994; Ferguson and Stentz, 2005; Gonzalez *et al.*, 2006; Stentz and Naggy, 2007]. The plan of the grounding result with the highest probability is executed by the robot. For more technical details on this work, we refer the reader to the longer version of this paper [Boularias *et al.*, 2015] and the intelligence architecture paper [Oh *et al.*, 2015] that describes how symbol grounding is used in semantic navigation.

2 Related Work

The challenge of building human-robot interfaces using natural language generated a large body of work [Harnad, 1990; MacMahon *et al.*, 2006; Matuszek *et al.*, 2012b; Zender *et al.*, 2009; Dzifcak *et al.*, 2009; Golland *et al.*, 2010; Tellex *et al.*, 2011; Kollar *et al.*, 2010; Tellex *et al.*, 2012; Walter *et al.*, 2013; Matuszek *et al.*, 2012a; Guadarrama *et al.*, 2013]. A full review of the related works is beyond the scope of this paper, so we highlight here some relevant examples. Symbol grounding was first formulated in [Harnad, 1990] as the problem of mapping words (symbols) into manifestations in the physical world. The Generalized Grounding Graphs (G^3) [Tellex *et al.*, 2011] is a generic framework that casts symbol grounding as a learning and inference problem in a Conditional Random Field. The same type of spatial relation clauses presented in [Kollar *et al.*, 2010] and used in [Tellex *et al.*, 2011] are used in the current work. The navigation system described in [Walter *et al.*, 2013], also based on G^3, incorporates odometry and path constraints in grounding, which is conceptually comparable to our use of perception confidence and path costs in grounding. Guadarrama et al. [Guadarrama *et al.*, 2013] presented a system for human-robot interaction that also learns both models for spatial prepositions and for object recognition. However, simple relations were considered and perception uncertainty was not taken into account.

3 Tactical Behavior Specification Grammar

The Tactical Behavior Specification (TBS) language is defined to instruct a robot to perform tactical behavior including navigation, searching for an object or observation. The language is specifically focused on describing desired behavior using spatial relationships with objects in an environment. In this paper, we focus on the navigate action, where the main components of a command are a goal and a navigation mode. An object (or a symbol) referenced in a command can be associated with a spatial constraint relative to other objects. For instance, in a command "Navigate covertly to a fire hydrant behind the building," a goal is to reach a fire hydrant, "behind the building" is a goal constraint, and "covertly" is the navigation mode. Often, the navigation mode also refers to an object. For instance, the navigation mode in "Navigate around the car to a fire hydrant behind the building" refers to an object named "car," which can also have its own spatial constraints that are independent from the constraints of the goal named "fire hydrant."

4 Navigation Mode Grounding

Once the landmarks have been grounded and the position of the goal has been determined, the robot must plan a path from its current position to the given goal location that obeys the path constraints imposed in the command. Path constraints describe a navigation mode. For example, the user may specify that the robot should stay "left of the building," or navigate "covertly". We return to the object grounding question in Section 5: "Which building should the robot stay left of?"

Path constraints are subjective and explicitly writing down a cost function that encapsulates them would be time consuming due to the many trade-offs inherently present in the planning problem. For example, the robot must trade off path length with distance from the building. A covert navigation behavior may look different to different people. We instead use Imitation Learning to learn *how* to navigate between a start and end position using examples of desired behavior.

We treat understanding spatial language as learning a mapping from terms (such as "left of," "around," or "covertly") to a cost function c which can be used to generate a matrix of costs known as a cost map. A planner can then optimize to produce the minimum cost path under this cost function. Given term σ (such as "left of") in the command specifying the navigation mode, the robot solves the planning problem of finding the minimum cost path ξ^* under cost function c_σ:

$$\xi^* = \operatorname*{argmin}_{\xi \in \Xi} c_\sigma\left(\xi\right) = \operatorname*{argmin}_{\xi \in \Xi} w_\sigma^T \phi\left(\xi\right) \qquad (1)$$

where the set of valid paths is Ξ, and we assume that the cost function c_σ takes the form of a linear sum of features ϕ under weights w_σ. The features describe the shape of the path, the geometry of the landmark, and the relationship between the two [Tellex *et al.*, 2011]. We use imitation learning to learn the weights w_σ from a set of demonstrated paths $\{\hat{\xi}_i\}_1^N$.

To learn the weights w_σ, we minimize the difference between the cost of the expert's demonstrated path $\hat{\xi}$ and the minimum cost path under the current cost function:

$$\ell\left(w_\sigma, \hat{\xi}\right) = w_\sigma^T \phi(\hat{\xi}) - \min_{\xi \in \Xi} w_\sigma^T \phi\left(\xi\right) + \frac{\lambda}{2}\|w_\sigma\|^2 \qquad (2)$$

under our regularization parameter λ. The first term in Equation 2 is the cost of the demonstrated path under the current cost function, and the second term is the cost of the optimal path (again, under the current cost function). Note that we are omitting the loss-augmentation term for clarity. Ignoring regularization, we achieve zero loss when the cost function produces the expert's path. This loss is optimized using the sub-gradient technique [Ratliff *et al.*, 2006].

Figure 2 shows the learned cost function for the relation "left of," and Figure 3 shows the training examples and learned cost function for "covert" navigation, along with the minimum cost path for a given start and end. Note that we learned to avoid being in the center area between the rows of buildings, and the planner took a sharper path across.

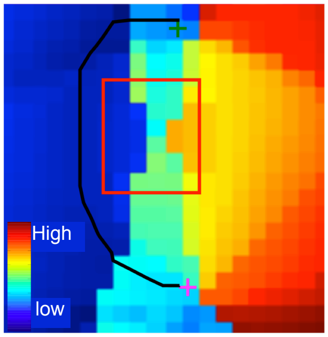
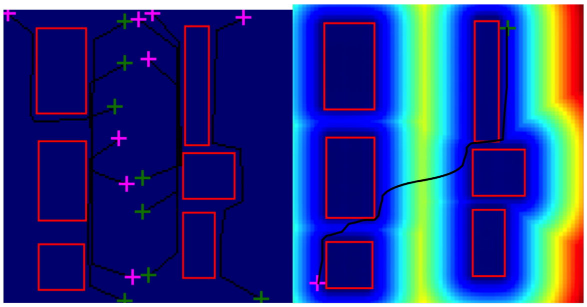


Figure 2: Learned cost function for the navigation mode "left of." The landmark is the rectangle in the center (*e.g.*, a building), and the path (optimized by minimizing the learned costs) correctly stays on the left side of the landmark.



(a) Demonstrated paths. (b) Learned cost function and a path generated accordingly.

Figure 3: Demonstrated paths and resulting learned cost function (along with a validation example) for "covert" navigation. Paths are drawn in black lines; buildings, red rectangles. Each path starts with a pink cross and ends with a green one.

5 Object Grounding With Spatial Constraints

5.1 Problem

The object grounding algorithm receives as inputs a text command, a set $\mathcal{O} = \{o_1, o_2 \ldots, o_n\}$ of perceived objects in the environment, and the position (x, y) of the robot at the time when the command was given. Each object o in set O is represented as a two-dimensional polygon, defined by the convex envelope of the object's points. Each object o is given a probability distribution P_o over labels $l \in \mathcal{L}$, obtained from the semantic perception module. For example, label l is "car" and $P_o(l)$ is the probability that object o is a car. A command contains one or more symbols from the label set $\mathcal{L}$. The symbols of particular interest for planning a path are the landmark-objects in goal and path constraints, denoted by ψ_g and ψ_p, respectively. For example, $\psi_g =$ "car" and $\psi_p =$ "building" in the command "Navigate near the building to the car that is behind the fire hydrant." The object grounding algorithm returns a joint probability distribution P on each pair of objects $(o_i, o_j) \in \mathcal{O} \times \mathcal{O}$. $P(o_i, o_j)$ is the probability that objects o_i and o_j are what the commander intended by symbols ψ_g and ψ_p, respectively. To compute $P(o_i, o_j)$, one needs to compute the probability of each object given all the symbols in the command, such as "behind" and "fire hydrant," in addition to symbols ψ_g and ψ_p. But only these two last symbols are used for planning a path. The other symbols only help grounding ψ_g and ψ_p. There are several ways for planning the robot's path based on the resulting distribution P. In this work, we pick the pair (o_i, o_j) that has the highest probability, and use it to generate a path according to the grounded navigation mode (Section 4). We show how P is computed in the rest of this section.

5.2 Model

Label prior distribution P_o is directly obtained from the object recognition method proposed in [Munoz, 2013]. A uniform distribution on all labels in $\mathcal{L}$ is used for objects that do not belong to any known class. This happens when the robot encounters a new type of objects for the first time. Also, the symbols in a received command that are not in $\mathcal{L}$ are automatically added to $\mathcal{L}$. P_o is adjusted such that the probabilities of the new labels are nonzero. For instance, the user refers to a "fire hydrant" in our previous example. That means that one of the objects in the environment has to be grounded as a 'fire hydrant' even if this term (or label) was never used before.

We use a log-linear model to represent $P_{\mathcal{R},o_i}(o_j)$, the probability that object $o_j \in \mathcal{O}$ is the one that satisfies spatial relation $\mathcal{R}$ with object $o_i \in \mathcal{O}$,

$$P_{\mathcal{R},o_i}(o_j) = \frac{\exp\left(w_{\mathcal{R}}^T \phi(x, y, o_i, o_j)\right)}{\sum_{o_k \in \mathcal{O}} \exp\left(w_{\mathcal{R}}^T \phi(x, y, o_i, o_k)\right)}, \tag{3}$$

wherein $\phi(x, y, o_i, o_j)$ is a vector of spatial features of the objects o_i and o_j from the robot's perspective at current position (x, y), and $w_{\mathcal{R}}$ is a vector of weights specific to relation $\mathcal{R}$. We dropped the robot's coordinates (x, y) and objects set $\mathcal{O}$ from the notation $P_{\mathcal{R},o_i}$ because they are constant during the grounding process. The spatial features used here are the distance between center(o_i) and center(o_j), the centers of objects o_i and o_j, in addition to the sine and cosine of the angle between (x, y)-center(o_i) axis and center(o_i)-center(o_j) axis. These features are adequate for learning spatial relations between relatively small objects. For large objects, such as buildings, the spatial relations depend on the overall shape and orientation of the object. Therefore, we use Principal Component Analysis (PCA) to find the primary and secondary axis of o_i when o_i is most likely a building (according to the perception module), and replace the (x, y)-center(o_i) axis by the nearest axis to it among the primary and secondary axis. We also define the distance between a building o_i and the center of another object o_j as the smallest of the distances between o_j and each vertex of o_i. These geometric features were sufficient for learning weights $w_{\mathcal{R}}$ of all the spatial relations used in our experiments. The same general approach can be used for learning other relations by using additional features.

5.3 Inference

We show how to compute a joint distribution P on landmarks named as ψ_g and ψ_p in the goal and the path constraints of a command. Each of the two objects can be subject to one or more spatial constraints, parsed as a binary tree and denoted by $\mathcal{T}_g$ and $\mathcal{T}_p$ respectively. We start by first computing a distribution on the objects in $\mathcal{O}$ for each label mentioned in the command. The object distribution, denoted by P_l for label l, is computed from the label distributions P_o (available from semantic perception) using Bayes' rule and a uniform prior. The next step consists in computing two distributions on goal and path landmarks, denoted as $P_{\mathcal{T}_g}$ and $P_{\mathcal{T}_p}$, from the spatial constraints in trees $\mathcal{T}_g$ and $\mathcal{T}_p$. The trees are traversed in a post-order depth-first search, which corresponds to reading the constraints in a reverse Polish notation. The logical operators ("and," "or," "not") are in the internal nodes of the tree, whereas the atomic spatial constraints ("behind building," "near car," etc.) are in the leaves.

5.4 Learning

Given a weight vector $w_{\mathcal{R}}$, probability $P_{\mathcal{R},o_i}(o_j)$ (Equation 3) indicates how likely a human user would choose o_j among all objects in a set $\mathcal{O}$ as the one that satisfies $\mathcal{R}(o_i, o_j)$. Because of perception uncertainties, estimating $P_{\mathcal{R},o_i}(o_j)$ for each object o_j is more important than simply finding the object that most satisfies relation $\mathcal{R}$ with o_j.

We used twenty examples for learning the spatial relations $\mathcal{R} \in \{$"left", "right", "front", "behind","near","away"$\}$. Each example i contains a set of objects in a simulated environment, a position (x_i, y_j) of the robot, a command with spatial constraints, in addition to the best answer o_i^* according to a human teacher. Weight vector $w_{\mathcal{R}}$ of each relation $\mathcal{R}$ is obtained by maximizing the log-likelihood of all the training examples using gradient descent, with the l_1 regularization for sparsifying the weights [Bishop, 2006].

6 Experiments

6.1 Simulation experiments

These experiments are a study involving three uninformed human subjects. We created a world model with eleven objects: a building, two cars, six traffic cones and two unknown objects. We used five simple commands and five complex commands. Each command contains a navigation mode ("quickly" or "covertly") with a spatial constraint of the path, in addition to a spatial constraint of the goal. Complex commands contain additional goal constraints. Participants were separately asked to point to the goal they would choose for executing each command. The best answer, chosen by a majority vote, is compared to the robot's answer. Table 1 shows that the robot's answer matches with the best answer in 80% of the commands. A robot's answer is counted as valid if it matches the answer of at least one participant. All the grounded goals were valid in this study. We also report the consensus rate that is the percentage of commands where all the three participants agreed on one answer. The low rates of consensus clearly show the advantage of customized human-robot interfaces that can learn from users. For instance, one participant interpreted "front of a building" as the side where the cars were located. Similarly, we asked each participant to classify the robot's path as conform to the navigation mode (style) and constraints or as non-conform. The mode was classified as conform by the majority of the participants in only 60% of the commands. We noticed that the participants had all different definitions of what it means to navigate covertly.

	Simple	Complex
Best goal	80%	80%
Valid goal	100%	100%
Consensus	40%	20%
Best navigation mode	60%	60%
Valid navigation mode	100%	100%
Consensus	60%	60%

Table 1: Comparing the learned model to human subjects, using simple and complex commands. Navigation mode refers to style. Notice how low is the consensus among the subjects on the best answers, which are chosen by a majority vote.

6.2 Robot experiments

We performed extensive experiments using the robotic platform shown in Figure 1. The robot's environment contained mainly buildings, cars, traffic cones, fire hydrants, and a gas pump. We evaluated the performance of the learned grounding model in five different scenes. In each scene, we used five simple commands and five complex ones. The total number of test scenarios is then 50. In each test scenario, we select a goal and a navigation mode (style), send a command to the robot, and rate the planned path as a success if it matches the selected goal and mode, and as a failure otherwise. Overall, we notice that complex commands help finding the right goals because they are less ambiguous than simple commands (88±11 vs. 84±17 success rate).

7 Conclusion

To become useful teammates, robots will need to understand natural language commands given to them. This problem is highly challenging when the environment is unknown. Spatial navigation and relations are one type of subjective linguistic concepts that robots can learn from human users. Our approach to solving this problem uses imitation learning for learning navigation modes, and a probabilistic model for trading off perception uncertainties with spatial constraints. Empirical evaluations show that the human-robot interface built using the proposed approach is an efficient tool for commanding mobile robots.

Acknowledgment

This work was conducted in part through collaborative participation in the Robotics Consortium sponsored by the U.S Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement W911NF-10-2-0016, and in part by ONR under MURI grant "Reasoning in Reduced Information Spaces" (no. N00014-09-1-1052). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[Bishop, 2006] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[Boularias *et al.*, 2015] Abdeslam Boularias, Felix Duvallet, Jean Oh, and Anthony Stentz. Grounding spatial relations for outdoor robot navigation. In *IEEE International Conference on Robotics and Automation, ICRA 2015, Seattle, WA, USA, 26-30 May 2015*, 2015.

[Dzifcak *et al.*, 2009] Juraj Dzifcak, Matthias Scheutz, Chitta Baral, and Paul W. Schermerhorn. 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 *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4163–4168, 2009.

[Ferguson and Stentz, 2005] David Ferguson and Anthony Stentz. Field D*: an interpolation-based path planner and replanner. In *Proceedings of the International Symposium on Robotics Research (ISRR)*, October 2005.

[Golland *et al.*, 2010] Dave Golland, Percy Liang, and Dan Klein. A game-theoretic approach to generating spatial descriptions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 410–419, 2010.

[Gonzalez *et al.*, 2006] Juan Pablo Gonzalez, Bryan Nagy, and Anthony Stentz. The geometric path planner for navigating unmanned vehicles in dynamic environments. In *Proceedings of the 1st Joint Emergency Preparedness and Response and Robotic and Remote Systems*, 2006.

[Guadarrama *et al.*, 2013] Sergio Guadarrama, Lorenzo Riano, Dave Golland, Daniel Gouhring, Yangqing Jia, Dan Klein, Pieter Abbeel, and Trevor Darrell. Grounding spatial relations for human-robot interaction. In *Proceedings of the 26th IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 1640–1647, 2013.

[Harnad, 1990] Stevan Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.

[Kollar *et al.*, 2010] Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas Roy. Toward understanding natural language directions. In *Proceedings of the 5th ACM/IEEE International Conference on Human-robot Interaction (HRI)*, pages 259–266, 2010.

[MacMahon *et al.*, 2006] Matt MacMahon, Brian Stankiewicz, and Benjamin Kuipers. Walk the Talk: Connecting Language, Knowledge, and Action in Route Instructions. In *National Conference on Artificial Intelligence*, 2006.

[Matuszek *et al.*, 2012a] Cynthia Matuszek, Nicholas Fitzgerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. A joint model of language and perception for grounded attribute learning. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 1671–1678, 2012.

[Matuszek *et al.*, 2012b] Cynthia Matuszek, Evan Herbst, Luke Zettlemoyer, and Dieter Fox. Learning to Parse Natural Language Commands to a Robot Control System. In *International Symposium on Experimental Robotics*, 2012.

[Munoz *et al.*, 2010] Daniel Munoz, J. Andrew Bagnell, and Martial Hebert. Stacked Hierarchical Labeling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.

[Munoz, 2013] Daniel Munoz. *Inference Machines: Parsing Scenes via Iterated Predictions*. PhD thesis, The Robotics Institute, Carnegie Mellon University, June 2013.

[Oh *et al.*, 2015] Jean Oh, Arne Suppe, Felix Duvallet, Abdeslam Boularias, Jerry Vinokurov, Luis Navarro-Serment, Oscar Romero, Robert Dean, Christian Lebiere, Martial Hebert, and Anthony Stentz. Toward Mobile Robots Reasoning Like Humans. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, 2015.

[Ratliff *et al.*, 2006] Nathan D. Ratliff, J. Andrew Bagnell, and Martin A. Zinkevich. Maximum Margin Planning. In *Proceedings of the International Conference on Machine Learning*, 2006.

[Stentz and Naggy, 2007] Anthony Stentz and Bryan Naggy. *PMAP User's Guide*. National Robotics Engineering Center, Carnegie Mellon University, 1.0 edition, Mar. 2007.

[Stentz, 1994] Anthony Stentz. Optimal and efficient path planning for partially-known environments. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 3310–3317, 1994.

[Tellex *et al.*, 2011] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R Walter, Ashis Gopal Banerjee, Seth J Teller, and Nicholas Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, 2011.

[Tellex *et al.*, 2012] Stefanie Tellex, Pratiksha Thaker, Robin Deits, Thomas Kollar, and Nicholas Roy. Toward information theoretic human-robot dialog. In *Robotics: Science and Systems IIX*, 2012.

[Walter *et al.*, 2013] Matthew R. Walter, Sachithra Hemachandra, Bianca Homberg, Stefanie Tellex, and Seth J. Teller. Learning semantic maps from natural language descriptions. In *Robotics: Science and Systems IX*, 2013.

[Zender *et al.*, 2009] Hendrik Zender, Geert-Jan M. Kruijff, and Ivana Kruijff-Korbayová. Situated resolution and generation of spatial referring expressions for robotic assistants. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1604–1609, 2009.