Toward Learning Perceptually Grounded Word Meanings from Unaligned Parallel Data

Stefanie Tellex and **Pratiksha Thaker** and **Josh Joseph** and **Matthew R. Walter** and **Nicholas Roy**
MIT Computer Science and Artificial Intelligence Laboratory

Abstract

In order for robots to effectively understand natural language commands, they must be able to acquire a large vocabulary of meaning representations that can be mapped to perceptual features in the external world. Previous approaches to learning these *grounded* meaning representations require detailed annotations at training time. In this paper, we present an approach which is capable of jointly learning a policy for following natural language commands such as "Pick up the tire pallet," as well as a mapping between specific phrases in the language and aspects of the external world; for example the mapping between the words "the tire pallet" and a specific object in the environment. We assume the action policy takes a parametric form that factors based on the structure of the language, based on the G^3 framework and use stochastic gradient ascent to optimize policy parameters. Our preliminary evaluation demonstrates the effectiveness of the model on a corpus of "pick up" commands given to a robotic forklift by untrained users.

1 Introduction

In order for robots to robustly understand human language, they must have access to meaning representations capable of mapping between symbols in the language and aspects of the external world which are accessible via the robot's perception system. Previous approaches have represented word meanings as symbols in some specific symbolic language, either programmed by hand [Winograd, 1971, MacMahon et al., 2006] or learned [Matuszek et al., 2010, Chen and Mooney, 2011, Liang et al., 2011, Branavan et al., 2009]. Because word meanings are represented as symbols, rather than perceptually grounded features, the mapping between these symbols and the external world must still be defined. Furthermore, the uncertainty of the mapping between constituents in the language and aspects of the external world cannot be explicitly represented by the model.

Language grounding approaches, in contrast, map words in the language to *groundings* in the external world [Mavridis and Roy, 2006, Hsiao et al., 2008, Kollar et al., 2010, Tellex et al., 2011]. Groundings are the specific physical concept that is referred to by the language and can be objects (e.g., a truck or a door), places (e.g., a particular location in the world), paths (e.g., a trajectory through the environment), or events (e.g., a sequence of robot actions). This symbol grounding approach [Harnad, 1990] represents word meanings as *functions* which take as input a perceptual representation of a grounding and return whether it matches words in the language. Recent work has demonstrated how to learn grounded word meanings from a parallel corpus of natural language commands paired with groundings in the external world [Tellex et al., 2011]. However, learning model parameters required that the parallel corpus be augmented with additional annotations specifying the alignment between specific phrases in the language and corresponding groundings in the external world. Figure 1 shows an example command from the training set paired with these alignment annotations, represented as arrows

Workshop on Semantic Interpretation in an Actionable Context, NAACL-HLT 2012, pages 7–14,
Montréal, Canada, June 3-8, 2012. ©2012 Association for Computational Linguistics

pointing from each linguistic constituent to a corresponding grounding.

Our approach in this paper relaxes these annotation requirements and learns perceptually grounded word meanings from an *unaligned* parallel corpus that only provides supervision for the top-level action that corresponds to a natural language command. Our system takes as input a state/action space for the robot defining a space of possible groundings and available actions in the external world. In addition it requires a corpus of natural language commands paired with the correct action executed in the environment. For example, an entry in the corpus consists of a natural language command such as "Pick up the tire pallet" given to a robotic forklift, paired with an action sequence of the robot as drives to the tire pallet, inserts its forks, and raises it off the ground, drives to the truck, and sets it down.

To learn from an unaligned corpus, we derive a new training algorithm that combines the Generalized Grounding Graph (G^3) framework introduced by Tellex et al. [2011] with the policy gradient method described by Branavan et al. [2009]. We assume a specific parametric form for the action policy that is defined by the linguistic structure of the natural language command. The system learns a policy parameters that maximize expected reward using stochastic gradient ascent. By factoring the policy according to the structure of language, we can propagate the error signal to each term, allowing the system to infer groundings for each linguistic constituent even without direct supervision. We evaluate our model using a corpus of natural language commands collected from untrained users on the internet, commanding the robot to pick up objects or drive to locations in the environment. The evaluation demonstrates that the model is able to predict both robot actions and noun phrase groundings with high accuracy, despite having no direct supervision for noun phrase groundings.

2 Background

We briefly review the G^3 framework, introduced by Tellex et al. [2011]. In order for a robot to understand natural language, it must be able to map between words in the language and corresponding groundings in the external world. The aim is to find

Figure 1: Sample entry from an aligned corpus, where mappings between phrases in the language and groundings in the external world are explicitly specified as arrows. Learning the meaning of "the truck" and "the pallet" is challenging when alignment annotations are not known.

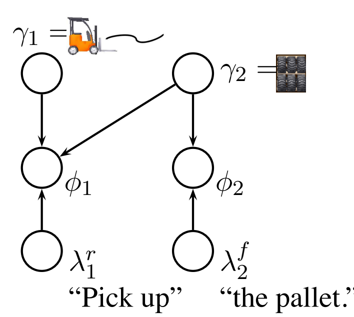


Figure 2: Grounding graph for "Pick up the tire pallet.

the most probable groundings $\gamma_1 \dots \gamma_N$ given the language Λ and the robot's model of the environment M:

$$\underset{\gamma_1 \cdots \gamma_N}{\operatorname{argmax}}\ p(\gamma_1 \dots \gamma_N | \Lambda, M) \tag{1}$$

M consists of the robot's location, the locations, geometries, and perceptual tags of objects, and available actions the robot can take. For brevity, we omit M from future equations in this section.

To learn this distribution, one standard approach is to factor it based on certain independence assumptions, then train models for each factor. Natural language has a well-known compositional, hierarchical argument structure [Jackendoff, 1983], and a promising approach is to exploit this structure in order to factor the model. However, if we define a directed model over these variables, we must assume a possibly arbitrary order to the conditional γ_i factors. For example, for a phrase such as "the tire pallet near the other skid," we could factorize in either of the following ways:

$$p(\gamma_{\text{tires}}, \gamma_{\text{skid}} | \Lambda) = p(\gamma_{\text{skid}} | \gamma_{\text{tires}}, \Lambda) \times p(\gamma_{\text{tires}} | \Lambda) \tag{2}$$

$$p(\gamma_{\text{tires}}, \gamma_{\text{skid}} | \Lambda) = p(\gamma_{\text{tires}} | \gamma_{\text{skid}}, \Lambda) \times p(\gamma_{\text{skid}} | \Lambda) \tag{3}$$

Depending on the order of factorization, we will need different conditional probability tables that correspond to the meanings of words in the language. To resolve this issue, another approach is to use Bayes' Rule to estimate the $p(\Lambda | \gamma_1 \dots \gamma_N)$, but this approach would require normalizing over all possible words in the language Λ. Another alternative is to use an undirected model, but this approach is intractable because it requires normalizing over all possible values of all γ_i variables in the model, including continuous attributes such as location and size.

To address these problems, the G^3 framework introduced a correspondence vector Φ to capture the dependency between $\gamma_1 \dots \gamma_N$ and Λ. Each entry in $\phi_i \in \Phi$ corresponds to whether linguistic constituent $\lambda_i \in \Lambda$ corresponds to grounding γ_i. We assume that $\gamma_1 \dots \gamma_N$ are independent of Λ *unless* Φ is known. Introducing Φ enables factorization according to the structure of language with local normalization at each factor over a space of just the two possible values for ϕ_i.

2.1 Inference

In order to use the G^3 framework for inference, we want to infer the groundings $\gamma_1 \dots \gamma_N$ that maximize the distribution

$$\underset{\gamma_1 \cdots \gamma_N}{\operatorname{argmax}}\ p(\gamma_1 \dots \gamma_N | \Phi, \Lambda) \tag{4}$$

which is equivalent to maximizing the joint distribution of all groundings $\gamma_1 \dots \gamma_N$, Φ and Λ,

$$\underset{\gamma_1 \cdots \gamma_N}{\operatorname{argmax}}\ p(\gamma_1 \dots \gamma_N, \Phi, \Lambda). \tag{5}$$

We assume that Λ and $\gamma_1 \dots \gamma_N$ are independent when Φ is not known, yielding:

$$\underset{\gamma_1 \cdots \gamma_N}{\operatorname{argmax}}\ p(\Phi | \Lambda, \gamma_1 \dots \gamma_N) p(\Lambda) p(\gamma_1 \dots \gamma_N) \tag{6}$$

This independence assumption may seem unintuitive, but it is justified because the correspondence variable Φ breaks the dependency between Λ and $\gamma_1 \dots \gamma_N$. If we do not know whether $\gamma_1 \dots \gamma_N$ correspond to Λ, we assume that the language does not tell us anything about the groundings.

Finally, for simplicity, we assume that any object in the environment is equally likely to be referenced by the language, which amounts to a constant prior on $\gamma_1 \dots \gamma_N$. In the future, we plan to incorporate models of attention and salience into this prior. We ignore $p(\Lambda)$ since it does not depend on $\gamma_1 \dots \gamma_N$, leading to:

$$\underset{\gamma_1 \cdots \gamma_N}{\operatorname{argmax}}\ p(\Phi | \Lambda, \gamma_1 \dots \gamma_N) \tag{7}$$

To compute the maximum value of the objective in Equation 7, the system performs beam search over $\gamma_1 \dots \gamma_N$, computing the probability of each assignment from Equation 7 to find the maximum probability assignment. Although we are using $p(\Phi | \Lambda, \gamma_1 \dots \gamma_N)$ as the objective function, Φ is fixed, and the $\gamma_1 \dots \gamma_N$ are unknown. This approach is valid because, given our independence assumptions, $p(\Phi | \Lambda, \gamma_1 \dots \gamma_N)$ corresponds to the joint distribution over all the variables given in Equation 5.

In order to perform beam search, we factor the model according to the hierarchical, compositional linguistic structure of the command:

$$p(\Phi | \Lambda, \gamma_1 \dots \gamma_N) = \prod_i p(\phi_i | \lambda_i, \gamma_{i_1} \dots \gamma_{i_k}) \tag{8}$$

This factorization can be represented graphically; we call the resulting graphical model the *grounding graph* for a natural language command. The directed model for the command "Pick up the pallet" appears in Figure 2. The λ variables correspond to language; the γ variables correspond to groundings in the external world, and the ϕ variables are $True$ if the groundings correspond to the language, and $False$ otherwise.

In the fully supervised case, we fit model parameters Θ using an aligned parallel corpus of labeled positive and negative examples for each linguistic constituent. The G^3 framework assumes a log-linear parametrization with feature functions f_j and feature weights θ_j:

$$p(\Phi|\Lambda, \gamma_1 \dots \gamma_N) = \tag{9}$$

$$\prod_i \frac{1}{Z} \exp(\sum_j \theta_j f_j(\phi_i, \lambda_i, \gamma_{i_1} \dots \gamma_{i_k})) \tag{10}$$

This function is convex and can be optimized with gradient-based methods [McCallum, 2002].

Features correspond to the degree to which each Γ correctly grounds λ_i. For a relation such as "on," a natural feature is whether the grounding corresponding to the head noun phrase is supported by the grounding corresponding to the argument noun phrases. However, the feature $supports(\gamma_i, \gamma_j)$ alone is not enough to enable the model to learn that "on" corresponds to $supports(\gamma_i, \gamma_j)$. Instead we need a feature that also takes into account the word "on:"

$$supports(\gamma_i, \gamma_j) \wedge (\text{"on"} \in \lambda_i) \tag{11}$$

3 Approach

Our goal is to learn model parameters for the G^3 framework from an unaligned corpus of natural language commands paired with robot actions. Previously, the system learned model parameters Θ using an aligned corpus in which values for all grounding variables are known at training time, and annotators provided both positive and negative examples for each factor. In this paper we describe how to relax this annotation requirement so that only the top-level action needs to be observed in order to train the model. It is easy and fast to collect data with these annotations, whereas annotating the values of all the variables, including negative examples is time-consuming and error prone. Once we know the model parameters we can use existing inference to find groundings corresponding to word meanings, as in Equation 4.

We are given a corpus of D training examples. Each example d consists of a natural language command Λ^d with an associated grounding graph with grounding variables Γ^d. Values for the grounding variables are not known, except for an observed value g_a^d for the top-level action random variable, γ_a^d. Finally, an example has an associated environmental context or semantic map M^d. Each environmental context will have a different set of objects and available actions for the robot. For example, one training example might contain a command given in an environment with a single tire pallet; another might contain a command given in an environment with two box pallets and a truck.

We define a sampling distribution to choose values for the Γ variables in the model using the G^3 framework with parameters Θ:

$$p(\Gamma^d|\Phi^d, \Lambda^d, M^d, \Theta) \tag{12}$$

Next, we define a reward function for choosing the correct grounding g_a for a training example:

$$r(\Gamma^d, g_a^d) = \begin{cases} 1 & \text{if } \gamma_a^d = g_a^d \\ -1 & \text{otherwise} \end{cases} \tag{13}$$

Here γ_a is a grounding variable for the top-level action corresponding to the command; it is one of the variables in the vector Γ. Our aim is to find model parameters that maximize expected reward when drawing values for Γ^d from $p(\Gamma^d|\Phi^d, \Lambda^d, M^d, \Theta)$ over the training set:

$$\operatorname*{argmax}_{\Theta} \sum_d E_{p(\Gamma^d|\Phi^d,\Lambda^d,M^d,\Theta)} r(\Gamma^d, g_a^d) \tag{14}$$

Expanding the expectation we have:

$$\operatorname*{argmax}_{\Theta} \sum_d \sum_{\Gamma}^{d} r(\Gamma^d, g_a^d) p(\Gamma^d|\Phi^d, \Lambda^d, M^d, \Theta) \tag{15}$$

We use stochastic gradient descent to find model parameters that maximize reward. First, we take the derivative of the expectation with respect to Θ. (We drop the d subscripts for brevity.)

$$\frac{\partial}{\partial \theta_k} E_{p(\Gamma|\Phi,\Lambda,M,\Theta)} r(\Gamma, g_a) = \sum_{\Gamma} r(\Gamma, g_a) \frac{\partial}{\partial \theta_k} p(\Gamma|\Phi, \Lambda, M, \Theta) \tag{16}$$

Focusing on the inner term, we expand it with Bayes' rule:

$$\frac{\partial}{\partial \theta_k} p(\Gamma|\Phi, \Lambda, M, \Theta) = \frac{\partial}{\partial \theta_k} \frac{p(\Phi|\Gamma, \Lambda, M, \Theta) p(\Gamma|\Lambda, M, \Theta)}{p(\Phi|\Lambda, M, \Theta)} \tag{17}$$

We assume the priors do not depend on Θ:

$$\frac{p(\Gamma|M)}{p(\Phi|\Lambda)} \frac{\partial}{\partial \theta_k} p(\Phi|\Gamma, \Lambda, M, \Theta) \tag{18}$$

For brevity, we compress Γ, Λ, and M in the variable X. Next, we take the partial derivative of the likelihood of Φ. First we assume each factor is independent.

$$\frac{\partial}{\partial \theta_k} p(\Phi|X, \Theta) = \frac{\partial}{\partial \theta_k} \prod_i p(\phi_i|X, \Theta) \tag{19}$$

$$= \prod_i p(\phi_i|X, \Theta) \times \left(\sum_j \frac{\frac{\partial}{\partial \theta_k} p(\phi_j|X, \Theta)}{p(\phi_j|X, \Theta)} \right) \tag{20}$$

Finally, we assume the distribution over ϕ_j takes a log-linear form with feature functions f_k and parameters θ_k, as in the G^3 framework.

$$\frac{\partial}{\partial \theta_k} p(\phi_j|X, \Theta) = \tag{21}$$

$$p(\phi|X, \Theta) \times \left(f_k(\phi, X) - E_{p(\phi'|X,\Theta)} \left[f_k(\phi', X) \right] \right)$$

We substitute back into the overall expression for the partial derivative of the expectation:

$$\frac{\partial}{\partial \theta_k} E_{p(\Gamma|X,\Theta)} r(\Gamma, g_a) = E_{p(\Gamma|X,\Theta)} r(\Gamma, g_a) \times \left(\sum_j f_k(\phi_j, X) - E_{p(\phi'|X,\Theta)} \left[f_k(\phi', X) \right] \right) \tag{22}$$

Input:
Initial values for parameters, Θ^0.
Training dataset, D.
Number of iterations, T.
Step size, α.

$\Theta \leftarrow \Theta^0$
for $t \in T$ **do**
  for $d \in D$ **do**
    $\nabla_\Theta \leftarrow \frac{\partial}{\partial \theta_k} E_{p(\Gamma^d|\Phi^d,\Lambda^d,M^d,\Theta)} r(\Gamma^d, g_a^d)$
  end for
  $\Theta \leftarrow \Theta + \alpha \nabla_\Theta$
end for
Output: Estimate of parameters Θ

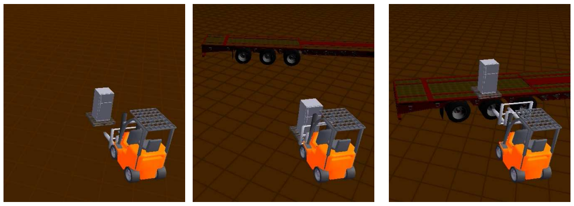
Figure 3: Training algorithm.

We approximate the expectation over Γ with highly probable bindings for the Γ and update the gradient incrementally for each example. The training algorithm is given in Figure 3.

4 Results

We present preliminary results for the learning algorithm using a corpus of natural language commands given to a robotic forklift. We collected a corpus of natural language commands paired with robot actions by showing annotators on Amazon Mechanical Turk a video of the robot executing an action and asking them to describe in words they would use to command an expert human operator to carry out the commands in the video. Frames from a video in our corpus, together with commands for that video appear in Figure 4. Since commands often spanned multiple parts of the video, we annotated the alignment between each top-level clause in the command and the robot's motion in the video. Our initial evaluation uses only commands from the corpus that contain the words "pick up" due to scaling issues when running on the entire corpus.

We report results using a random cost function as a baseline as well as the learned parameters on a training set and a held-out test set. Table 1 shows performance on a training set using small environments (with one or two other objects) and a test set of small and large environments (with up to six other objects).



(a) t=0 (b) t=20 (c) t=30

```
Pick up pallet with refridgerator [sic] and place on truck to
the left.

A distance away you should see a rectangular box.  Approach it
slowly and load it up onto your forklift.  Slowly proceed to
back out and then make a sharp turn and approach the truck.
Raise your forklift and drop the rectangular box on the back of
the truck.

Go to the pallet with the refrigerator on it and pick it
up.  Move the pallet to the truck trailer.  Place the pallet on
the trailer.

Pick up the pallet with the refrigerator and place it on
the trailer.
```

(d) Commands

Figure 4: Frames from a video in our dataset, paired with natural language commands.

	% Correct	
	Actions	Concrete Noun Phrases
Before Training	31%	61%
After Training	100%	92%

(a) Training (small environments)

	% Correct	
	Actions	Concrete Noun Phrases
Before Training	3%	26%
After Training	84%	77%

(b) Testing (small and large environments)

Table 1: Results on the training set and test set.

As expected, on the training set the system learns a good policy, since it is directly rewarded for acting correctly. Because the environments are small, the chance of correctly grounding concrete noun phrases with a random cost function is high. However after training performance at grounding noun phrases increases to 92% even though the system had no access to alignment annotations for noun phrases at training time; it only observes reward based on whether it has acted correctly.

Next, we report performance on a test set to assess generalization to novel commands given in novel environments. Since the test set includes larger environments with up to six objects, baseline performance is lower. However the trained system is able to achieve high performance at both inferring correct actions as well as correct object groundings, despite having no access to a reward signal of any kind during inference. This result shows the system has learned general word meanings that apply in novel contexts not seen at training time.

5 Related Work

Beginning with SHRDLU [Winograd, 1971], many systems have exploited the compositional structure of language to statically generate a plan corresponding to a natural language command [Hsiao et al., 2008, MacMahon et al., 2006, Skubic et al., 2004, Dzifcak et al., 2009]. Our work moves beyond this framework by defining a probabilistic graphical model according to the structure of the natural language command, inducing a distribution over plans and groundings.

Models that learned word meanings [Tellex et al., 2011, Kollar et al., 2010] require detailed alignment annotations between constituents in the language and objects, places, paths, or events in the external world. Previous approaches capable of learning from unaligned data [Vogel and Jurafsky, 2010, Branavan et al., 2009] used sequential models that could not capture the hierarchical structure of language. Matuszek et al. [2010], Liang et al. [2011] and Chen and Mooney [2011] describe models that learn compositional semantics, but word meanings are symbolic structures rather than patterns of features in the external world.

There has been a variety of work in transferring action policies between a human and a robot. In imitation learning, the goal is to create a system that can watch a teacher perform an action, and then reproduce that action [Kruger et al., 2007, Chernova and Veloso, 2009, Schaal et al., 2003, Ekvall and Kragic, 2008]. Rybski et al. [2007] developed an imitation learning system that learns from a combination of imitation of the human teacher, as well as natural language input. Our work differs in that the system must infer an action from the natural language commands, rather than from watching the teacher perform an action. The system is trained off-line, and the task of the robot is to respond on-line to the natural language command.

6 Conclusion

In this paper we described an approach for learning perceptually grounded word meanings from an unaligned parallel corpus of language paired with robot actions. The training algorithm jointly infers policies that correspond to natural language commands as well as alignments between noun phrases in the command and groundings in the external world. In addition, our approach learns grounded word meanings or distributions corresponding to words in the language, that the system can use to follow novel commands that it may have never encountered during training. We presented a preliminary evaluation on a small corpus, demonstrating that the system is able to infer meanings for concrete noun phrases despite having no direct supervision for these values.

There are many directions for improvement. We plan to train our system using a large dataset of language paired with robot actions in more complex environments, and on more than one robotic platform. Our approach points the way towards a framework that can learn a large vocabulary of general grounded word meanings, enabling systems that flexibly respond to a wide variety of natural language commands given by untrained users.

References

S. R. K. Branavan, H. Chen, L. S. Zettlemoyer, and R. Barzilay. Reinforcement learning for mapping instructions to actions. In *Proceedings of ACL*, page 82–90, 2009.

D. L. Chen and R. J. Mooney. Learning to interpret natural language navigation instructions from observations. In *Proc. AAAI*, 2011.

S. Chernova and M. Veloso. Interactive policy learning through confidence-based autonomy. *JAIR*, 34 (1):1–25, 2009.

J. Dzifcak, M. Scheutz, C. Baral, and P. 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 *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, pages 4163–4168, 2009.

S. Ekvall and D. Kragic. Robot learning from demonstration: a task-level planning approach. *International Journal of Advanced Robotic Systems*, 5(3), 2008.

S. Harnad. The symbol grounding problem. *Physica D*, 43:335–346, 1990.

K. Hsiao, S. Tellex, S. Vosoughi, R. Kubat, and D. Roy. Object schemas for grounding language in a responsive robot. *Connection Science*, 20(4): 253–276, 2008.

R. S. Jackendoff. *Semantics and Cognition*, pages 161–187. MIT Press, 1983.

T. Kollar, S. Tellex, D. Roy, and N. Roy. Toward understanding natural language directions. In *Proc. ACM/IEEE Int'l Conf. on Human-Robot Interaction (HRI)*, pages 259–266, 2010.

V. Kruger, D. Kragic, A. Ude, and C. Geib. The meaning of action: A review on action recognition and mapping. *Advanced Robotics*, 21(13), 2007.

P. Liang, M. I. Jordan, and D. Klein. Learning dependency-based compositional semantics. In *Proc. Association for Computational Linguistics (ACL)*, 2011.

M. MacMahon, B. Stankiewicz, and B. Kuipers. Walk the talk: Connecting language, knowledge, and action in route instructions. In *Proc. Nat'l Conf. on Artificial Intelligence (AAAI)*, pages 1475–1482, 2006.

C. Matuszek, D. Fox, and K. Koscher. Following directions using statistical machine translation. In *Proc. ACM/IEEE Int'l Conf. on Human-Robot Interaction (HRI)*, pages 251–258, 2010.

N. Mavridis and D. Roy. Grounded situation models for robots: Where words and percepts meet. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4690–4697. IEEE, Oct. 2006. ISBN 1-4244-0258-1.

A. K. McCallum. MALLET: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

P. Rybski, K. Yoon, J. Stolarz, and M. Veloso. Interactive robot task training through dialog and demonstration. In *Proceedings of HRI*, page 56. ACM, 2007.

S. Schaal, A. Ijspeert, and A. Billard. Computational approaches to motor learning by imitation. *Phil. Trans. R. Soc. Lond. B*, (358), 2003.

M. Skubic, D. Perzanowski, S. Blisard, A. Schultz, W. Adams, M. Bugajska, and D. Brock. Spatial language for human-robot dialogs. *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34(2):154–167, 2004. ISSN 1094-6977.

S. Tellex, T. Kollar, S. Dickerson, M. Walter, A. Banerjee, S. Teller, and N. Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proc. AAAI*, 2011.

A. Vogel and D. Jurafsky. Learning to follow navigational directions. In *Proc. Association for Computational Linguistics (ACL)*, pages 806–814, 2010.

T. Winograd. *Procedures as a Representation for Data in a Computer Program for Understanding Natural Language*. PhD thesis, Massachusetts Institute of Technology, 1971. Ph.D. thesis.