Approaching the Symbol Grounding Problem with Probabilistic Graphical Models

Stefanie Tellex[1] and **Thomas Kollar**[1] and **Steven Dickerson** and
Matthew R. Walter and **Ashis Gopal Banerjee** and **Seth Teller** and **Nicholas Roy**
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139

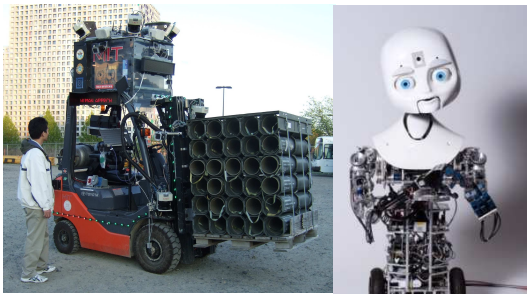
Abstract

In order for robots to engage in dialog with human teammates, they must have the ability to identify correspondences between elements of language and aspects of the external world. A solution to this symbol grounding problem (Harnad, 1990) would enable a robot to interpret commands such as "Drive over to receiving and pick up the tire pallet." This article describes several of our results that use probabilistic inference to address the symbol grounding problem. Our approach is to develop models that factor according to the linguistic structure of a command. We first describe an early result, a generative model that factors according to the sequential structure of language, then discuss our new framework, Generalized Grounding Graphs (G^3). The G^3 framework dynamically instantiates a probabilistic graphical model for a natural language input, enabling a mapping between words in language and concrete objects, places, paths and events in the external world. We report on corpus-based experiments in which the robot is able to learn and use word meanings in three real-world tasks: indoor navigation, spatial language video retrieval, and mobile manipulation.

1 Introduction

As robots move out of the lab and into the real world, it is critical to develop ways for human users to easily and flexibly command them. Natural language dialog is a compelling solution to this problem because the operator can flexibly express complex requirements, enabling interaction as if it were another human. In order to effectively engage in dialog, a robot must be able to interpret natural language commands. For example, a human supervisor might tell an autonomous forklift, "Put the tire pallet on the truck" (Figure 1a), or an operator might command a humanoid robot, "Drive down the hall past the elevators." (Figure 1b).

A critical component to understanding commands like these is ability to map words in the language to aspects of the external world. This mapping, which Harnad (1990) called the symbol grounding problem, has been studied since the early days of artificial intelligence. There are broadly three different ways people have approached the symbol grounding problem in robotics. Starting with Winograd (1970),



Pick up the pallet of boxes in the middle and place them on the trailer to the left.

(a) Robotic Forklift.

Go down the hall past the elevators to the kitchen.

(b) Humanoid Robot.

Figure 1: Target robotic platforms and example mobile manipulation and navigation commands.

many have manually created symbol systems that map between language and the external world, connecting each term onto a pre-specified action space and set of environmental features (Bugmann et al., 2004; Dzifcak et al., 2009; Hsiao, Mavridis, and Roy, 2003; Kress-Gazit and Fainekos, 2008; MacMahon, Stankiewicz, and Kuipers, 2006; Roy, Hsiao, and Mavridis, 2003; Roy, 2005). This class of systems takes advantage of the structure of spatial language, but usually do not involve learning, have little perceptual feedback, and have a fixed action space. A second approach involves learning the meaning of words in the sensorimotor space (e.g., joint angles and images) of the robot (Marocco et al., 2010; Modayil and Kuipers, 2007; Sugita and Tani, 2005). By treating linguistic terms as a sensory input, these systems must learn directly from complex features extracted by perceptual systems, resulting in a limited set of commands that they can robustly understand. A third approach is to use learning to convert from language onto aspects of the environment. These approaches may use only linguistic features (Ge and Mooney, 2005; Shimizu and Haas, 2009), spatial features (Regier, 1992) or linguistic, spatial and semantic features (Branavan et al., 2009; Branavan, Silver, and Barzilay, 2011; Kollar et al., 2010b; Matuszek, Fox, and Koscher, 2010; Vogel and Jurafsky, 2010). These approaches learn the meaning of spatial prepositions (e.g.,

[1]The first two authors contributed equally to this paper.

"above" Regier 1992), verbs of manipulation (e.g., "push" and "shove" Bailey 1997), and verbs of motion (e.g., "follow" and "meet" Kollar et al. 2010a) and landmarks (e.g., "the doors" Kollar et al. 2010b).

In this paper, we give an overview of our probabilistic approach to the symbol grounding problem. By taking a probabilistic approach, we are able to build systems that learn word meanings from large corpora of examples and use those meanings to find good groundings in the external world, despite uncertainty. Our first approach uses a generative model that factors according to the sequential structure of language. This model can be used to follow natural language route instructions and to perform spatial language video retrieval. However, the generative approach requires explicit corpora for each modeled factor, rather than learning word meanings directly from in-domain language. It can not represent complex linguistic structures such as referring expressions (e.g., "the door across from the elevators") and multi-argument verbs (e.g., "put the pallet on the truck.") To address these limitations, we developed a new framework, called Generalized Grounding Graphs (G^3), introduced in Tellex et al. (2011). The G^3 framework dynamically instantiates a conditional probabilistic graphical model that factors according to the compositional and hierarchical structure of a natural language phrase. Using the new model, we created a system that successfully follows many mobile-manipulation commands from a corpus created by untrained annotators using crowd sourcing.

Several earlier publications describe the primary technical contributions of the models (Kollar et al., 2010a,b; Tellex et al., 2010, 2011). This paper provides an integrated overview of results and lessons learned for the generative model and the G^3 framework in three domains: navigation in indoor environments, spatial language video retrieval, and mobile manipulation.

2 Approach

Our goal is a framework that can map between language and the external world. We assume a natural language utterence Λ has a corresponding set of *groundings*, Γ, in the external world. Groundings 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), and events (e.g., a sequence of robot actions). We additionally assume a semantic map m, consisting of the locations and labels of other objects in the environment. The distribution we want to model is:

$$p(\Lambda, \Gamma, m) \tag{1}$$

We explicitly represent the joint distribution rather than the conditional because it can be used to solve several types of problems. To interpret commands, one can optimize over candidate groundings; this paper describes our work in this area. Furthermore, the model could be used to generate natural language descriptions of objects or events by searching for for a description Λ. Finally it could also be used to recognize events by directly computing the probability of a particular set of values and thresholding. The challenge in making this approach practical for any of these problems is factoring the distribution and providing models for each factor.

2.1 Generative Model

In our previous work (Kollar et al., 2010b), we approached this problem by factoring Equation 1 to give $p(\Lambda|\Gamma, m) \times p(\Gamma, m)$. This formulation allowed us to make independence assumptions corresponding to the sequential clause structure of the language, yielding:

$$p(\Lambda|\Gamma, m) = \prod_i p(\lambda_i|\Gamma, m) \tag{2}$$

where λ_i are the words associated with each clause. We assumed that each factor had a fixed structure: a verb v, a spatial relation, sr, and a landmark, l. Furthermore, we assumed that each clause had a fixed set of groundings, consisting of a path fragment, p, and an object, o:

$$p(\lambda_i|\Gamma, m) = p(v, sr, l|p, o, m) \tag{3}$$

We then made independence assumptions based on this structure.

$$p(v, sr, l|p, o, m) = p(v|p) \times p(sr|p, o) \times p(l|p, m) \tag{4}$$

This approach allowed us to define individual models for each term in the factorization. We defined models for pre-specified verbs and adverbs such as "left," "right," and "straight," and we trained models for spatial relations such as "to," "past," and "through." For the landmark factor, we exploited co-occurrence statistics from a large online database of labeled images (Flickr) to estimate the probability of an unknown landmark phrase given objects detected in the semantic map (Kollar and Roy, 2009). These statistics enabled the robot to estimate the probability of seeing a landmark phrase such as "the kitchen" using a limited set of existing object detectors, such as a refrigerator and a sink.

We tested our generative model in two real-world domains: following natural language directions through real-world environments and spatial language video retrieval. The model can be used to follow natural language directions by finding the path through the environment, $\gamma_{path} \in \Gamma$, that maximizes the distribution in Equation 3. To evaluate the system at following natural language directions, we collected from 15 subjects a corpus of 150 directions through a large office environment. Our system successfully followed 67% of the directions in the corpus, compared to human performance of 85%.

For video retrieval, the task was to find video clips from a large corpus (Roy et al., 2006) that match a spatial language description of a person's motion, such as "Show me people walking into the kitchen." The system performed ranked retrieval by scoring video clips according to how well they matched a spatial language query according to Equation 3. We demonstrated that our system could effectively retrieve video clips, evaluating on a large corpus of natural language queries created by untrained users. Figure 2 shows a sample query result for the system.

However, the generative framework has several limitations. First, because it only models the flat sequential structure of language, rather than the hierarchical structure, it

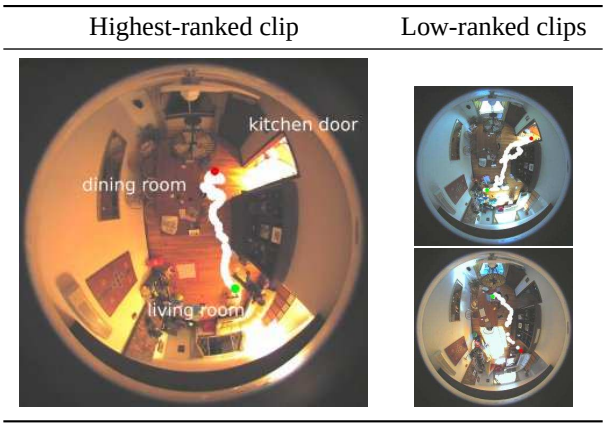


Figure 2: Results from the generative model (Tellex et al., 2010) for the query "from the couches in the living room to the dining room table." The person's start location is marked with a green dot; the end location is marked with a red dot, and their trajectory is marked in white.

cannot handle commands such as "Go to the door across from the elevators." The phrase "the door across from the elevators" is treated as a bag of words, and the system is unable to distinguish whether to approach the door or the elevators. Second, we assumed that each clause has a fixed structure consisting of a path and a landmark, but language has variable, hierarchical structure. The flat structure cannot support two-argument verbs like "Put the tire pallet on the truck," or nested arguments. Third, it is difficult to obtain models for the meanings of words in the individual factors. In our route directions dataset, people often used "thru" instead of "through." Since the system learned word meanings from a separate, curated corpus that did not contain "thru," we manually encoded this synonymy. The system was unable to learn word meanings directly from the corpus.

2.2 Generalized Grounding Graphs

Our aim in creating the G^3 framework was to address the challenges from the previous section by modeling the hierarchical, compositional structure of language in a framework that could learn word meanings from data. To facilitate learning, we converted the distribution in Equation 1 to a discriminative model (Kollar, Tellex, and Roy, 2010) by introducing a correspondence vector, Φ:

$$p(\Phi|\Gamma, \Lambda, m) \tag{5}$$

The correspondence vector Φ contains a boolean variable ϕ for each linguistic constituent $\lambda \in \Lambda$ and corresponding grounding $\gamma \in \Gamma$, such that ϕ is true if λ and γ correspond and false otherwise.

The G^3 framework factors the model according to the structure of the language, allowing explicit inference over groundings for each linguistic constituent:

$$p(\Phi|\Lambda, \Gamma, m) = \prod_i p(\phi_i|\lambda_i, \Gamma, m) \tag{6}$$

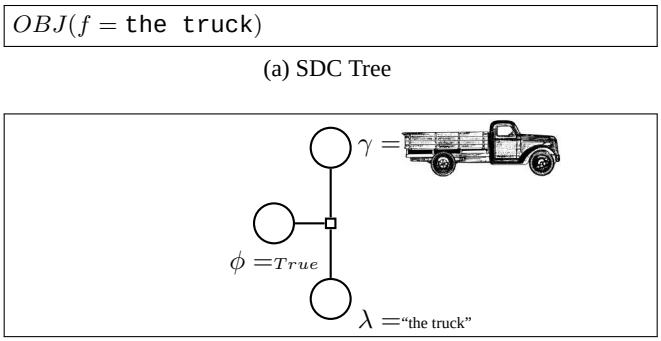


Figure 3: SDCs and grounding graph for the phrase "the truck" showing one set of values for the variables. This graph corresponds to the probability distribution $p(\Phi|\Lambda, \Gamma, m) = p(\phi|\gamma, \lambda, m)$.

This factorization can be represented graphically as a *factor graph* (Kschischang, Frey, and Loeliger, 2001). A factor graph is a bipartite graph with two types of nodes: random variables and factors. Each factor node corresponds to a factor in the distribution and connects to variable nodes, which are its arguments. For example, Figure 3 shows a factor graph for the phrase "the truck," consisting of a single factor and three variables: γ, which is a vector of features corresponding to an object in the external world with a particular appearance and location, λ, the words "the truck," and ϕ, which is true if γ corresponds to λ, and false otherwise. The graph corresponds to the distribution $p(\phi|\gamma, \lambda, m)$. (If λ in Figure 3 were the words "the tire pallet," then $p(\phi = False|\gamma, \lambda)$ would have higher probability.) Since the semantic map m appears in all factors, we omit it from the graphical representation. We refer to factor graphs created by the G^3 framework as *grounding graphs*. Word models in each factor can be learned discriminatively, and the resulting factorization allows the system to compose them in order to follow novel commands that may have never been seen in training.

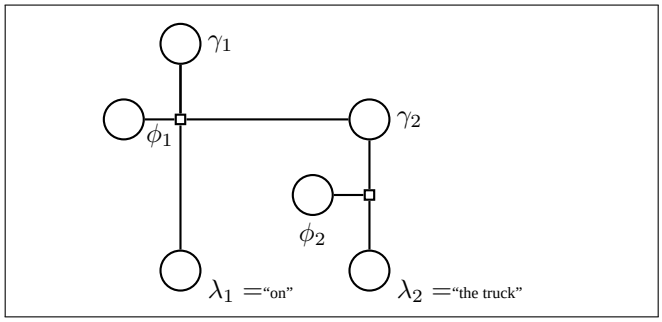
In order to precisely define the factorization in Equation 6, we use Spatial Description Clauses (SDCs). SDCs were introduced by Kollar et al. (2010b) and refined by Tellex et al. (2011); they correspond to the parse structure of a natural language command. An SDC consists of a *figure* phrase f, a *relation* r, and a variable number of *landmark* noun phrases l_i. We assign a type to each SDC following the system defined by Jackendoff (1983):

- **EVENT** Something that takes place (or should take place) in the world (e.g., "Move the tire pallet" or "Turn right").
- **OBJECT** A thing in the world. This category includes people and the robot as well as physical objects (e.g., "Forklift," "the tire pallet," "the hallway," "the person").
- **PLACE** Places in the world (e.g., "on the truck," "next to the tire pallet" or "in the kitchen").
- **PATH** Paths through the world (e.g., "past the truck," or "down the hall").

SDCs with relations contain one or more arguments.

$PLACE_2(r = \texttt{on}$
$l1 = OBJ_1(f = \texttt{the truck}))$

(a) SDC Tree



(b) Induced Model

Figure 4: SDCs and grounding graph for the phrase "on the truck." This graph corresponds to the probability distribution $p(\Phi|\Lambda, \Gamma, m) = p(\phi_1|\lambda_1, \gamma_1, \gamma_2, m) \times p(\phi_2|\lambda_2, \gamma_2, m)$.

Since almost all relations take two core arguments or less, we use at most two landmark fields l_1 and l_2. Given this definition, a general natural language command is represented as a sequence of SDC trees. An SDC tree for the command "Put the pallet on the truck" appears in Figure 5a. Leaf SDCs in the tree contain only text in the figure field, such as "the truck" (Figure 3a). Internal SDCs contain text in the relation field and child SDCs in the figure and landmark fields.

The system automatically extracts SDCs from the Stanford dependency parse structure(de Marneffe, MacCartney, and Manning, 2006). The SDC extraction algorithm maps between particular dependency types and fields in the SDCs, putting verbs and prepositions in the relation field, their arguments in the landmark field, and their subjects in the figure field. In cases of ambiguity the algorithm outputs multiple candidate SDCs for a single parse. We obtain additional candidates by running the extractor on the n-best list of parse candidates. The system then performs discriminative reranking using a model trained from annotated SDCs.

Using SDCs, we can rewrite the inner term from Equation 6 as:

$$p(\phi_i|\lambda_i, \Gamma, m) = p(\phi_i|SDC_i, \Gamma, m) \tag{7}$$

Further independence assumptions can be made in the product terms based on the structure of the language. To specify these factors, we first define the variables in the model as follows:

- ϕ_i True if the grounding γ_i corresponds to i^{th} SDC.
- λ_i^f The text of the figure field of the i^{th} SDC.
- λ_i^r The text of the relation field of the i^{th} SDC.
- $\gamma_i^f, \gamma_i^{l1}, \gamma_i^{l2} \in \Gamma$ The groundings associated with the corresponding field of the i^{th} SDC: the robot or object state sequence, or a location in the semantic map.

Looking at Equation 7, we can see that the model has a factor for each SDC in the parse. The dynamically generated factors fall into two types:

- $p(\phi_i|\lambda_i^f, \gamma_i, m)$ for leaf SDCs.
- $p(\phi_i|\lambda_i^r, \gamma_i^f, \gamma_i^{l1}, m)$ or $p(\phi_i|\lambda_i^r, \gamma_i^f, \gamma_i^{l1}, \gamma_i^{l2}, m)$ for internal SDCs.

Leaf factors always correspond to an OBJECT or PLACE SDC and operate over the correspondence variable ϕ_i, the figure text λ_i^f and a unique grounding γ_i. An internal factor corresponds to an OBJECT, PLACE, PATH, or EVENT SDC which has text in the relation field. The arguments to these factors are the correspondence variable ϕ_i, relation text λ_i^r, and the candidate groundings γ_i^f and γ_i^{l1} (and optionally γ_i^{l2}) corresponding to the figure and landmark fields of an SDC.

For example, Figure 4 shows the grounding graph for the phrase "on the truck." It contains a subgraph corresponding to "the truck" which is identical to the one shown in Figure 3. The value of the correspondence variable ϕ_1 depends only on the values of λ_1 ("on") and the groundings γ_1 (a place in the world) and γ_2 (an object), and not on the specific words "the truck." This independence assumption enables the model to represent a general meaning for "on" that does not depend on specific text in its argument phrase.

Each factor in Equation 7 is a log-linear model with the following form (Lafferty, McCallum, and Pereira, 2001):

$$p(\phi_i|\text{SDC}_i, \Gamma, m) = \frac{1}{Z} \exp\left(\sum_k \mu_k s_k(\phi_i, \text{SDC}_i, \Gamma, m)\right) \tag{8}$$

Here, s_k are feature functions (described more fully in Tellex et al. 2011) that take as input a correspondence variable, an SDC and a set of groundings and output a binary value. For example, one of the many feature functions corresponds to whether the landmark grounding γ_i^l is supporting the figure grounding γ_i^f and the word "on" is in the relation field of the SDC:

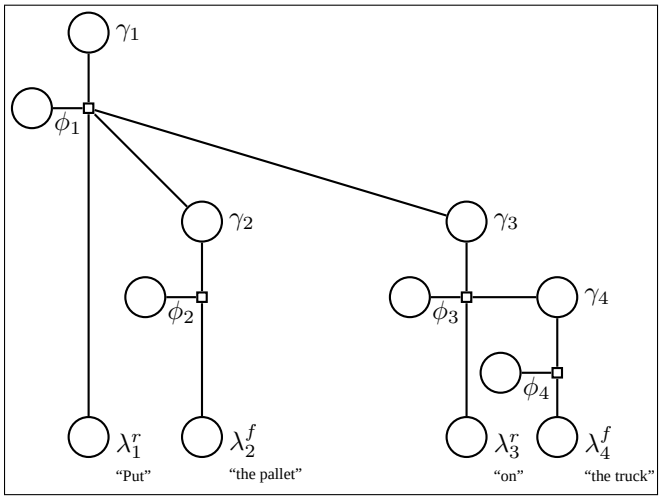
$$f(\gamma_i^f, \gamma_i^l, \lambda_i^r) \equiv supports(\gamma_i^f, \gamma_i^l) \wedge (\text{"on"} \in \lambda_i^r) \tag{9}$$

We use features relating the distance between the figure and the landmark groundings, as well as the change in state at the beginning and end of the robot's trajectory. Features are created based on the syntactic role of the words in the language: whether it appears as a figure, relation, or landmark in the SDC. To ground noun phrases, the system assumes access to an object detector that can recognize certain classes of objects, such as pallets and trucks. The system learns to map between these labels and words that actually appeared in the command, such as "skid" or "trailer." We also use features derived from co-occurrence statistics from large web corpora, such as Flickr, as described by Kollar et al. (2010b).

The μ_k are the weights corresponding to the output of a particular feature function. At training time, we observe SDCs, groundings Γ, and the output vector Φ. In order to learn the parameters μ_k that maximize the likelihood of the

$$EVENT_1(r = \texttt{Put},\ l = OBJ_2(f = \texttt{the pallet}),\ l2 = PLACE_3(r = \texttt{on},\ l = OBJ_4(f = \texttt{the truck})))$$

(a) SDC tree



(b) Induced Model

$$p(\Phi|\Gamma,\text{SDCs},m) = p(\phi_1|\gamma_1,\gamma_2,\gamma_3,\lambda_1^r = \text{Put},m)\times p(\phi_2|\gamma_2,\lambda_2^f = \text{the pallet},m)\times p(\phi_3|\gamma_3,\gamma_4,\lambda_3^r = \text{on},m)\times p(\phi_4|\gamma_4,\lambda_4^f = \text{the truck},m)$$

(c) Factorization

Figure 5: In (a) is SDC tree for "Put the pallet on the truck." In (b) is the induced graphical model and in (c) is the factorization.

training dataset, we use L-BFGS (Andrew and Gao, 2007) to optimize the parameters of the model via gradient descent.

Figure 5 shows an entire worked example for the command "Put the pallet on the truck," beginning with SDCs, the grounding graph, and finally the factorization of the distribution. Note that the factor graph contains subgraphs corresponding to the constituents "on the truck" (shown in Figure 4) and "the truck" (shown in Figure 3). This decomposition allows the model to learn word meanings from each factor and flexibly compose them together in order to understand novel commands.

3 Results

We present results from experiments with the G^3 framework using three corpora of natural language commands paired with robot actions and environment state sequences. Examples from the corpora appear in Figure 6. We used one part of these corpora to train the G^3 model to learn the meanings of words and used a held-out test set to evaluate the end-to-end performance of the system at composing word meanings in order to follow commands.

The first corpus focuses on spatial prepositions describing paths, such as "across," "to," "toward," and "along." Each example in the corpus consists of a trajectory, a landmark object, and a phrase such as "Go to the door" or "Go across the conference room;" the corpus includes both positive and negative examples of each spatial relation. One of the authors created the corpus by drawing a sequence of waypoints that corresponded to a phrase such as "down the hallway." Negative examples were created by treating positive examples of one spatial relation as negative examples of another, with some exceptions such as "to" and "toward." This dataset provides a simple test bed to demonstrate the model's performance, as well as providing training examples for bootstrapping the model on this important class of words. Figure 6a shows a sample prepositional phrase from this corpus, paired with a path and landmark.

The second corpus consists of natural language route instructions. We collected a corpus of 150 natural language route instructions from fifteen people, through one floor of two adjoining office buildings. An example set of directions from the corpus is shown in Figure 6b. Following these directions is challenging because they consist of natural language constrained only by the task and as a result may use any of the complicated linguistic structures associated with free-form natural language. This corpus provides a complex sample of spatial language for a real-world task. To train the model, we annotated each constituent in the corpus with a corresponding path segment or landmark. We constructed negative examples by randomizing these annotations. Figure 6b shows a sample command from the corpus.

The third corpus consists of mobile-manipulation commands given to a robotic forklift. Annotators on Amazon Mechanical Turk watched a video of a simulated forklift performing an action, then wrote natural language commands they would give to an expert human operator in order to command them to carry out the actions in the video. This corpus consists of a rich variety of mobile-manipulation commands such as "Pick up the pallet of tires directly in front of the forklift." Figure 6c shows an example command from this dataset.

3.1 Meanings For Words

Next, we trained models for each of the corpora and evaluated their performance for specific words in a held-out test set, using the same features for all models and annotated parses. Table 1a shows the performance on words from the spatial relations corpus. Not surprisingly, it learned good models for the meanings of words in this simple corpus. To illustrate the learned models for individual words, we present the probability distribution as a heat map, where red is high probability and blue is low probability. Figure 7 shows maps for "to the truck," "past the truck" and "toward the truck," demonstrating that the system has learned nuanced models for these different words.

Table 1b shows the performance of the trained system on individual examples from the route directions corpus. Performance is lower because this corpus contained fewer examples of individual spatial relations and was noisier in gen-

(a) Spatial Relations. (b) Route Directions. (c) Mobile Manipulation.

Figure 6: Commands paired with environments from corpora used in our experiments.

Word	F-score	Accuracy	# of examples
path prepositions:			
across	0.77	0.83	42
around	1.00	1.00	218
past	0.71	0.98	218
through	0.75	0.83	24
to	0.93	0.99	474
toward	0.84	0.99	214

(a) Spatial Relations.

Word	F-score	Accuracy	# of examples
path prepositions:			
across	0.75	0.75	8
around	0.80	0.80	10
past	0.80	0.83	30
through	0.81	0.81	114
to	0.72	0.71	144
toward	0.61	0.69	29
place prepositions:			
near	1.00	1.00	8
on	0.98	0.98	55
verbs:			
take	0.92	0.93	40

(b) Route Directions.

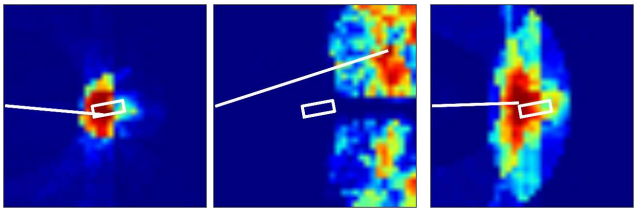
Word	F-score	Accuracy	# of examples
path prepositions:			
to	0.78	0.79	48
toward	0.80	0.75	4
place prepositions:			
near	0.00	0.50	4
on	0.66	0.66	62
verbs:			
lift	0.88	0.87	60
put	1.00	1.00	6
take	1.00	1.00	12

(c) Mobile Manipulation.

Table 1: Performance of the learned model in terms of recognizing the right actions for various words (i.e., correctly predicting ϕ). The final column shows the number of examples in the test set, with a 70%-30% training-testing split.

eral. The effects of this noise can be seen in the heat map shown in Figure 9b.

Finally, Table 1c shows the performance of the system when trained on the mobile manipulation corpus. The system was able to learn good models for verbs such as "put" and "take" as well as spatial relations such as "to," "toward" and "on" from relatively few training examples.



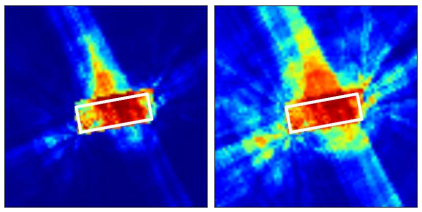
(a) "to the truck" (b) "past the truck" (c) "toward the truck"

Figure 7: Heat maps showing high and low probability ending locations for various phrases according to our learned model trained on the spatial relations dataset. The path is constrained to be a straight line starting at the left edge of the image. The highest probability path is drawn in white.

The word "take" appeared in both the mobile manipulation corpus and the route directions corpus, but it was used in different ways. In the route directions corpus, it was used in phrases such as "Take your first left," while in the mobile manipulation corpus, it was used in commands like "Take the pallet of tires to the trailer on the left." Although the system learned these two senses separately, learning from a single corpus that contained both would be challenging because the same feature weights would be trained for both word senses simultaneously.

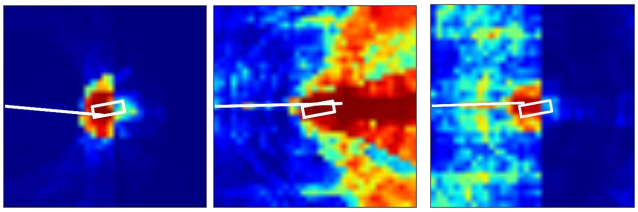
Figure 8a shows the distribution of locations for "on" as learned from the mobile manipulation corpus from phrases such as "put the pallet on the truck." (The target locations are a constant height above the ground.) The system gives high weight to locations that are supported by the truck, because features related to "support" have the highest weight among the learned features for "on." Figure 8b shows the distributions for the phrase "near the truck," which is not as peaked as strongly as "on." The distributions are asymmetric with respect to the truck because of frame-of-reference features which take into account the position and orientation of the robot. We intended these features to capture phrases like "on your left" and "to the left of," but the system also weights them for "on" and "near."

Figure 9 shows maps for "to the truck" from models trained on each of the three datasets. The system is able to learn good models from both the spatial relation and mobile manipulation datasets. The mobile manipulation dataset



(a) "on the truck." (b) "near the truck."

Figure 8: Heat map showing high probability (red) and low probability (blue) locations for "on the truck" and "near the truck" according to our learned model trained on the mobile manipulation corpus. The location of the truck is drawn in white.



(a) Spatial relations. (b) Route directions. (c) Mobile manipulation.

Figure 9: Heat maps showing high and low probability ending locations for a path corresponding to "to the truck." The path is constrained to be a straight line starting at the left edge of the image. The highest probability path is drawn in white.

is noisier because it contains fewer training examples, and many of the examples were part of compound prepositional phrases such as "to the left of the truck." The route instructions corpus is biased to go past the landmark object, probably because examples of "to" often occurred in the context of longer phrases such as "walk to the end of the hall and turn left."

3.2 End-to-end Evaluation

The fact that the model performed well at predicting the correspondence variable from annotated SDCs and groundings is promising but does not necessarily translate to good end-to-end performance when using the model to follow natural language commands.

To assess end-to-end performance, we evaluated the system in the mobile manipulation domain as described by Tellex et al. (2011). For each command in the corpus, the system inferred a plan and executed it in a realistic robot simulator. Then, annotators ranked whether the robot's behavior was correct or incorrect given the command. By this metric, our system correctly followed 54% of the thirty most confident commands in the corpus. When using a ground-truth parse instead of an automatic parse, the system followed 47% of commands from the entire corpus, and 63% of the thirty most confident commands.

The system qualitatively produced compelling end-to-end performance. When the system did make mistakes, it was often partially correct. For example, it might pick up the left tire pallet instead of the right one. Other problems stemmed from ambiguous or unusual language in the corpus commands, such as "remove the goods" or "then swing to the right," that make the inference particularly challenging. Despite these limitations, however, the system successfully followed commands such as "put the tire pallet on the truck," "pick up the tire pallet" "put down the tire pallet" and "go to the truck," using only data from the corpus to learn the model.

An enabling technology for our approach to mobile-manipulation is the availability of infrastructure for reliably simulating and logging robot actions. We useed these technologies to collect corpora of language paired with robot actions to train the system. We were then able to simulate the robot and automatically produce videos of the system following each command in the corpus, which we used for the end-to-end evaluation.

4 Lessons Learned

An important next step is to leverage larger corpora of language paired with robot actions. Children hear millions of words in many different contexts as they acquire language. The relative lack of data was the cause of many of the errors our system made. For example, annotator referred to a pallet that was separated from other pallets as "the lonely pallet," but the word "lonely" did not appear in the training set. As a result, the system was unable to learn a model for this word. Our learning framework requires detailed alignment annotation between linguistic constituents and groundings in the world, which limits our ability to leverage larger datasets. Our next goal is to reduce the amount of annotation required by using algorithms that alternate between picking labels and learning models using the inferred labels. A second approach to this problem would be to acquire word meanings from existing large corpora (Kollar and Roy, 2009). The challenge here is to identify datasets that would allow the system to map from words such as "pick up" or "lonely" to actions and perceptual features accessible to the robot.

A second challenge is interpreting high-level commands such as "unload the truck" that might require long sequences of primitive actions, as well as low-level commands such as "drive forward six inches." An action space detailed enough to represent actions such as driving forward a small distance will require extremely long action sequences to generate behavior like unloading a truck. This problem was made concrete by one of our annotators, who posted instructions for picking up a dime with a forklift:

> Raise the forks 12 inches. Line up either fork in front of the dime. Tilt the forks forward 15 degrees. Pull the truck forward until one fork is directly over the dime. Completely lower the forks. Put the truck in reverse and gently travel backward a foot. The dime will flip up backwards onto the fork. Level the forks back to 90 degrees. Raise the dime with the forks 12 inches.

To handle different granularities of actions, we are develop-

ing a hierarchical action space and new search algorithms that will enable the robot to efficiently search among both large-scale and small-scale actions when following a command.

A third challenge is learning word meanings that generalize across different domains without retraining the model. Figure 9 shows three different meanings for the word "to" learned from three different datasets. A further challenge is modifying learned models in response to modifiers, such as "half-way to the truck." Modeling nuanced changes of meaning in different contexts remains a challenging problem.

The ability to understand spatial language discourse and engage in dialog is critical to enable robots to robustly interact with humans using language. The model described here represents an early step toward a framework for acquiring word meanings, but much remains to be done. A system that can understand the full complexity of language must be able to handle ellipsis (when words are omitted from sentences), conditional expressions (e.g., "if a truck comes in, unload it"), and quantifiers (e.g., "move all the tire pallets"). It must also reason about uncertainty from the speech recognizer about what the person actually said, as well as uncertainty in the parser, such as ambiguous prepositional phrase attachment. We envision a joint search over speech recognition candidates, parse structures, and groundings in the world, applying information from multiple modalities to jointly reduce uncertainty. Furthermore the system must be able to combine multiple utterances into higher-level semantic units. Finally, it must be embedded in a higher-level dialog understanding framework that can reason about the system's uncertainty and take actions to reduce it, such as asking questions. Grounding graphs provide a building block to address these problems, but a more sophisticated framework must be developed to utilize them effectively.

5 Conclusion

This paper describes our probabilistic approach to the symbol grounding problem. We first reviewed a generative model that factors according to the sequential structure of language. Next we presented a hierarchical model, called Generalized Grounding Graphs (G^3), that is able to learn word meanings from corpora and compose them to understand novel commands. We described applications of the G^3 framework to several different domains and presented results demonstrating that it has learned the meanings of complex spatial prepositions and verbs.

6 Acknowledgments

We would like to thank Dimitar Simeonov, Alejandro Perez, and Nick dePalma as well as the annotators on Amazon Mechanical Turk and the members of the Turker Nation forum. This work was sponsored by the Robotics Consortium of the U.S Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement W911NF-10-2-0016, and by the Office of Naval Research under MURI N00014-07-1-0749.

References

Andrew, G., and Gao, J. 2007. Scalable training of L1-regularized log-linear models. In *Proc. Int'l Conf. on Machine Learning (ICML)*.

Bailey, D. 1997. *When Push Comes to Shove: A Computational Model of the Role of Motor Control in the Acquisition of Action Verbs*. Ph.D. Dissertation.

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

Branavan, S.; Silver, D.; and Barzilay, R. 2011. Learning to win by reading manuals in a Monte-Carlo framework. In *Proceedings of ACL*.

Bugmann, G.; Klein, E.; Lauria, S.; and Kyriacou, T. 2004. Corpus-based robotics: A route instruction example. *Proceedings of Intelligent Autonomous Systems* 96—103.

de Marneffe, M.; MacCartney, B.; and Manning, C. 2006. Generating typed dependency parses from phrase structure parses. In *Proc. Int'l Conf. on Language Resources and Evaluation (LREC)*, 449–454.

Dzifcak, J.; Scheutz, M.; Baral, C.; and Schermerhorn, P. 2009. 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)*, 4163–4168.

Flickr. http://www.flickr.com.

Ge, R., and Mooney, R. J. 2005. A statistical semantic parser that integrates syntax and semantics. In *Proc. of the Ninth Conference on Computational Natural Language Learning*, 9–16. Stroudsburg, PA, USA: Association for Computational Linguistics.

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

Hsiao, K.-y.; Mavridis, N.; and Roy, D. 2003. Coupling perception and simulation: Steps towards conversational robotics. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1 of *IROS '03*, 928–933. IEEE.

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

Kollar, T., and Roy, N. 2009. Utilizing object-object and object-scene context when planning to find things. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, 4116–4121.

Kollar, T.; Tellex, S.; Roy, D.; and Roy, N. 2010a. Grounding verbs of motion in natural language commands to robots. In *Proceedings of the International Symposium on Experimental Robotics (ISER)*.

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

Kollar, T.; Tellex, S.; and Roy, N. 2010. A Discriminative Model for Understanding Natural Language Route Directions. *In Proc. AAAI Fall Symposium on Dialog with Robots*.

Kress-Gazit, H., and Fainekos, G. E. 2008. Translating structured English to robot controllers. *Advanced Robotics* 22:1343–1359.

Kschischang, F. R.; Frey, B. J.; and Loeliger, H.-A. 2001. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory* 47(2):498 –519.

Lafferty, J. D.; McCallum, A.; and Pereira, F. C. N. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. Int'l Conf. on Machine Learning (ICML)*, 282–289.

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

Marocco, D.; Cangelosi, A.; Fischer, K.; and Belpaeme, T. 2010. Grounding action words in the sensorimotor interaction with the world: experiments with a simulated iCub humanoid robot. *Frontiers in Neurorobotics* 4(0).

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

Modayil, J., and Kuipers, B. 2007. Autonomous development of a grounded object ontology by a learning robot. In *Proc. AAAI*, volume 2, 1095–1101. AAAI Press.

Regier, T. P. 1992. *The Acquisition of Lexical Semantics for Spatial Terms: A Connectionist Model of Perceptual Categorization*. Ph.D. Dissertation, University of California at Berkeley.

Roy, D.; Patel, R.; DeCamp, P.; Kubat, R.; Fleischman, M.; Roy, B.; Mavridis, N.; Tellex, S.; Salata, A.; Guinness, J.; Levit, M.; and Gorniak, P. 2006. The Human Speechome Project. In *Proceedings of the 28th Annual Cognitive Science Conference*, 192–196.

Roy, D.; Hsiao, K.; and Mavridis, N. 2003. Conversational Robots: Building blocks for grounding word meanings. *Proceedings of the HLT-NAACL03 workshop on learning word meaning from non-linguistic data*.

Roy, D. 2005. Semiotic schemas: A framework for grounding language in action and perception. *Artif. Intell.* 167(1-2):170–205.

Shimizu, N., and Haas, A. 2009. Learning to follow navigational route instructions. In *Proceedings of the 21st International Joint Conference on Artifical Intelligence*, 1488–1493.

Sugita, Y., and Tani, J. 2005. Learning semantic combinatoriality from the interaction between linguistic and behavioral processes. *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems* 13:33–52.

Tellex, S.; Kollar, T.; Shaw, G.; Roy, N.; and Roy, D. 2010. Grounding spatial language for video search. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction*, ICMI-MLMI '10, 31:1–31:8. ACM.

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

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

Winograd, T. 1970. *Procedures as a representation for data in a computer program for understanding natural language.* Ph.D. Dissertation, Massachusetts Institute of Technology.

7 Biographies

Stefanie Tellex is a Research Scientist at the MIT Computer Science and Artificial Intelligence Laboratory at MIT. As part of her Ph.D. thesis at the MIT Media Lab, Stefanie developed models for the meanings of spatial prepositions and motion verbs. She has presented her work at SIGIR, HRI, AAAI and ICMI, describing systems for searching surveillance video with spatial language queries and for giving instructions to mobile robots.

Thomas Kollar has a Ph.D. in Electrical Engineering and Computer Science (EECS) from the Massachusetts Institute of Technology (MIT). His thesis concerned learning to understand spatial natural language commands and his research interests include robot learning, language grounding, and human-robot interaction. He was the general chair of the HRI Pioneers Workshop at the 6th ACM/IEEE International Conference on Human-Robot Interaction. He received his Master of Science in EECS from MIT in 2007 for research in reinforcement learning toward improving the quality of robot mapping. He has a Bachelor of Science in Computer Science (with Honors) and a Bachelor of the Arts in Mathematics from the University of Rochester. As an undergraduate, he developed an hors d'oeuvre-serving robot as a part of the AAAI robotics competition and is a member of IEEE, AAAI, and Sigma Xi and has published at HRI, ISER, ICRA, AAAI, IROS and ICMI.

Steven Dickerson graduated from MIT in 2011 with undergraduate degrees in Computer Science and Aerospace Engineering. He is currently employed by Goldman Sachs.

Matthew R. Walter is a Research Scientist in the Computer Science and Artificial Intelligence Laboratory at MIT. He received his Ph. D. in Mechanical Engineering from the Massachusetts Institute of Technology and the Woods Hole Oceanographic Institution in 2008. His research interests include perception, motion planning, and human-robot interaction, so as to enable mobile robots to operate safely and effectively within unstructured environments.

Seth Teller is a Professor in the Department of Electrical Engineering and Computer Science, and a member of the Computer Science and Artificial Intelligence Laboratory, at MIT. He received his Ph. D. in Computer Science from U.C. Berkeley in 1992. Teller's research interests include machine perception, mobile manipulation, human-robot interaction and assistive technology.

Ashis Gopal Banerjee is a Postdoctoral Associate in the Computer Science and Artificial Intelligence Laboratory at MIT. He completed his Ph.D. in Mechanical Engineering at the University of Maryland (UMD) in 2009. Prior to that, he obtained his Masters Degree in Mechanical Engineering at UMD in 2006 and Bachelors Degree in Manufacturing Science and Engineering at IIT Kharagpur in 2004. He received the 2009 Best Dissertation Award from the Department of Mechanical Engineering and the 2009 George Harhalakis Outstanding Systems Engineering Graduate Student Award at UMD. His research interests include planning under uncertainty, machine learning, and micro and nano manipulation.

Nicholas Roy is an Associate Professor in the Department of Aeronautics & Astronautics at the Massachusetts Institute of Technology and a member of the Computer Science and Artificial Intelligence Laboratory at MIT. He received his Ph. D. in Robotics from Carnegie Mellon University in 2003. His research interests include mobile robotics, decision-making under uncertainty, human-computer interaction, and machine learning.