Grounding Language for Physical Systems
AAAI Technical Report WS-12-07

Learning to Parse and Ground Natural Language Commands to Robots

Jayant Krishnamurthy and **Thomas Kollar**

Abstract

This paper describes a weakly supervised approach for understanding natural language commands to robotic systems. Our approach, called the combinatory grounding graph (CGG), takes as input natural language commands paired with groundings and infers the space of parses that best describe how to ground the natural language command. The command is understood in a compositional way, generating a latent hierarchical parse tree that involves relations (such as "to" or "by") and categories (such as "the elevators" or "the doors"). We show an example parse-grounding tree and show that our system can successfully cluster the meanings of objects and locations.

Introduction

To be useful teammates to human partners, robots must be able to robustly follow spoken instructions. For example, a human that is interacting with an autonomous robot might say, "Go through the door near the elevators." Understanding these commands typically involves two parts: parsing and grounding, each of which are treated as separate steps (Kollar et al. 2010; **?**). During parsing, the robot learns to translate a natural language command into an abstract representation of its meaning. During grounding, the robot identifies the correspondence between the command's meaning and the real-world environment.

In this paper, we describe the combinatory grounding graph (CGG), which dynamically instantiates a probabilistic graphical model over parses and groundings. The parse tree is a latent variable in the graphical model, and training uses a weakly supervised approach that takes a corpus of natural language commands paired with their groundings to simultaneously learn (1) an appropriate meaning representation language for the domain, (2) a semantic parser which translates sentences into the meaning representation and (3) a grounding function which maps the meaning representation on to the real world. Learning directly from language/grounding pairs avoids laborious annotation of the intermediate meaning representation. We show example CGG parses as well as some preliminary experiments showing that the system is able to learn categories that generalize over similar types of groundings and similar ways to refer to the same grounding.

Approach

In order to understand a natural language command we construct an undirected probabilistic graphical model that is able to infer the most probable groundings Γ given a natural language command Z:

$$\arg\max_{\Gamma} p(\Gamma|Z) \tag{1}$$

Inferring groundings over arbitrary language Z is a challenging problem because the input language could be arbitrarily complex. We address this complexity by introducing an intermediate hidden semantic parse P, which represents the correspondence of words in the command to groundings in the physical environment:

$$p(\Gamma|Z) = \sum_{P} p(\Gamma, P|Z) \tag{2}$$

The semantic parse P is a meaning representation that abstracts over semantically identical language, thereby enabling generalization across different words in the language and groundings in the environment. Previous work (Kollar et al. 2010; **?**) has assumed that the semantic structure factors according to a fixed syntactic parse of the command; CGGs learn the distribution over the possible semantic structures that best predicts the groundings.

Combinatory Categorial Grammar

The CGG uses a probabilistic Combinatory Categorial Grammar (CCG) (Steedman 1996) to identify linguistic structures with the same essential meaning. The CCG transforms language into a formula in first-order logic by combining the meanings of individual words. These meanings are represented by lexical entries such as:

door $:= N : \lambda x.c(x)$
double $:= N/N : \lambda f.\lambda x.f(x) \wedge c(x)$
by $:= (N\backslash N)/N : \lambda f.\lambda g.\lambda x.g(x) \wedge \exists y.f(y) \wedge r(x, y)$

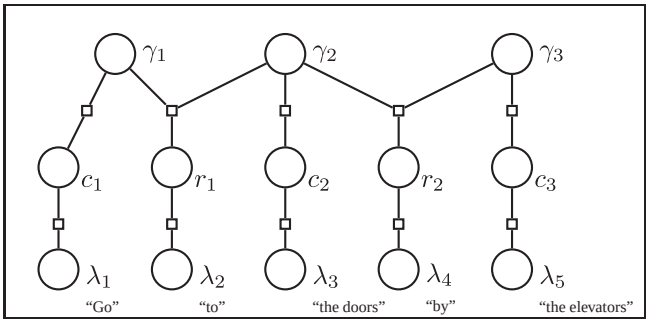
Each lexical entry maps a phrase to a syntactic type (e.g., N for noun), and a semantic type containing hidden categories c and relations r. Hidden categories c represent objects, places and people (e.g., "kitchen" and "computer") while relations r represent spatial relationships or actions (e.g., "next to", "within", "put", or "go to"). Such lexical entries can be imputed using part-of-speech heuristics. The CCG parser produces a set of logical forms for commands; an example parse of "Go through the door by the elevators," can be seen in Figure 1.

"Go through the door near the elevators."

(a) Natural Language Command

$\lambda x.\exists y, z.c_1(x) \wedge r_1(x, y) \wedge c_2(y) \wedge r_2(y, z) \wedge c_3(z)$
$\lambda x.\exists y, z.c_1(x) \wedge r_1(x, y) \wedge c_2(y) \wedge r_2(x, z) \wedge c_3(z)$

(b) Possible Logical Forms



(c) Factor graph representing $p(\Gamma, P|Z)$

Figure 1: In (a) is the natural language command. (b) shows several possible logical forms for this command. (c) shows the grounding graph generated from the topmost logical form. In this case γ_2, γ_3 are objects or locations and γ_1 is a path.

Parsing and Grounding

Given the command, a logical form and a set of groundings Γ from the environment, the probability of a particular parse/grounding pair can be computed. This probability decomposes according to the structure of the logical form into factors Ψ and Φ over random variables γ_i (the ith grounding), λ (the text) and r_i, c_j (the hidden relations, categories) in an undirected probabilistic graphical model:

$$p(\Gamma, P|Z) = \frac{1}{Z} \prod_i \Psi(\gamma_i, \gamma_j, r_i|\lambda) \prod_j \Phi(\gamma_j, c_j|\lambda) \quad (3)$$

Each factor is a log-linear model over a set of features. We decompose the learning into two components: (1) learning the mapping from text λ to a category c or relation r (e.g., weights $w_{c,\lambda}, w_{r,\lambda}$) and (2) learning the mapping from the category or relation onto groundings γ (e.g., weights $w_{g,r}, w_{g,c}$):

$$\Phi(c_i, \gamma_k|\lambda) = \exp\left(f(\gamma_i, c_i) \cdot w_{g,c} + f(c_i, \lambda) \cdot w_{c,\lambda}\right)$$
$$\Psi(r_j, \gamma_i, \gamma_k|\lambda) = \exp\left(f(\gamma_i, \gamma_k, r_j) \cdot w_{g,r} + f(r_j, \lambda) \cdot w_{r,\lambda}\right)$$

The learned parse parameters model how concepts and relations are referred to in language, while grounding parameters represent the mapping from concepts and relations to real-world objects, places, paths and events.

Grounding features were introduced in (Kollar et al. 2010) and Tellex 2011, and include a set of continuous features of two types: (1) geometric features of a fixed location of places and objects and (2) geometric features of the relationship between the path of an agent and the fixed location of objects and places. For example, relational geometric features are between two three-dimensional boxes in the world, such as the distance between two groundings $distance(\gamma_i, \gamma_j)$ for "near" and "by." The semantic parser uses lexical features, such as $count(\text{door} := N : \lambda x.c(x))$, which counts the number of times that "door" maps to the category $N : \lambda x.c(x)$ in the semantic parse.

Word	Syntax	Weight
hall	N	0.28
second	N/N	0.23
hallway	N	0.16
doorway	N	0.14

(a) word/category weights

Grounding Feature	Weight
groundingPerimeter	1.23
relatedTo_intersection	0.77
oftenSeenWith_cabinet	0.48
relatedTo_32d-886	0.36

(b) cat./grounding weights

Table 1: Highest-weight features for a learned category representing hallways. The weights in (a) represent words likely to map to the latent category during parsing. The synonymy between "hallway" and "hall" is learned and "second" and "doorway" are associated with the concept. The weights in (b) represent features of physical locations to which the category grounds; the perimeter of the grounding is the most likely predictor of "hallway," then the presence of an intersection and cabinets and (less highly weighted) a specific room (32d-886).

Examples and Preliminary Evaluation

An example is shown in Figure 1. At inference time, the CGG observes a natural language command and generates multiple different semantic representations (logical forms) for the command. It then searches for groundings that are consistent with each of these candidate semantic representations, weighting each grounding by the probability of the semantic representation. The grounding graph is generated from the structure of the logical form, which in turn is generated by composing the CCG lexical entries.

We have performed a preliminary evaluation of the system using a corpus of route directions from Kollar et al. (2010), taking only adjective/noun phrases (such as "double doors") referring to locations. The CGG learns to predict locations for noun phrases by inducing a meaning representation language containing 50 categories. Training uses stochastic gradient descent on the marginal log-likelihood, to simultaneously identify (1) semantic parser parameters mapping adjectives and nouns to these categories via lexical entries, and (2) grounding function parameters mapping each category to locations in the physical world. Table 1 shows parameters for a learned category from the meaning representation language, which can be interpreted as representing a hallway.

References

Kollar, T.; Tellex, S.; Roy, D.; and Roy, N. 2010. Toward understanding natural language directions. In *Proceedings of HRI-2010*.

Steedman, M. 1996. *Surface Structure and Interpretation*. The MIT Press.

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.