Interactive Learning and its Role in Pervasive Robotics

Cynthia Matuszek Dieter Fox Nicholas FitzGerald Evan Herbst

Abstract— **As robots have become lower-cost, more ubiquitous, and more capable, the importance of enabling untrained users to interact with them has increased. Such robots have the potential to provide assistance and reduce workloads in the home, in the workplace, and in the context of assistive technologies. However, it is difficult to predict the specific tasks that these robots should be programmed to assist with before they are deployed, and in these settings, robots will often be interacting with non-expert end users. In this paper, we argue that one approach to dealing with this type of human-robot interaction is *teachable robotics*, in which robots learn to perform novel tasks in novel environments from humans using intuitive teaching modalities, such as natural language. We describe two recent projects that make progress in this direction, and discuss the challenges revealed by this work.**

I. Introduction and Motivation

As robots have become more ubiquitous and capable, the importance of enabling untrained users to interact with them has increased. Small, low-cost robots have the potential to provide assistance and reduce workloads in the home, in the workplace, and in the context of assistive technologies. (Fig. 1 shows a robot designed for such tasks.)

However, despite this breadth of *possible* applications, it is not practical to predict every *specific* task that ubiquitous helper robots will need to perform. A manipulator robot might be asked to assist with soldering at a workbench, pipetting in a laboratory, or chopping an onion, depending on the context and the users' needs. One possible solution is to have end users themselves instruct robots about the world, including goals and actions, as needed for their particular situation. In this model, the focus is on the interactions between a human non-specialist and a *teachable robot,* a system able to accept and learn from instructions describing how to perform tasks.

Interactive learning is a broad problem, with components including natural language (NL) understanding, user interface design, active learning, learning by demonstration, gaze and gesture tracking, and probabilistic world modeling. In this paper, we discuss our work on *natural language grounding*–the interpretation of human natural language into semantically informed structures in the context of robotic perception and actuation.

Human instruction-giving is a rich area; modalities such as speech, gesture, gaze, and demonstration are all natural mechanisms by which humans teach, and learn from, one another. In response, unconstrained natural-language interaction with robots has emerged as a significant research area. The integration of natural language instruction with teaching by action and demonstration is intuitive and comfortable for human users, while offering sufficient signal to support robot task planning and any necessary modeling of previously unknown world state.

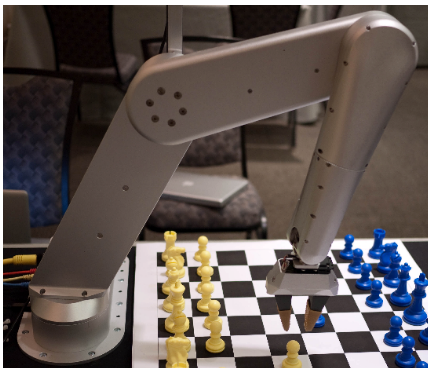


Fig. 1: The mid-cost Gambit manipulator arm was designed for small-scale tabletop manipulation tasks. Here, it plays chess against a human opponent.

At a high level, our goal is to make it easier for untrained users to interact with robots in a comfortable way. This requires robots with the ability to learn from natural language, in unfamiliar environments, about words and objects that the systems have not previously encountered. To this end, we use as case studies two projects we have undertaken. In both, natural language grounding is treated as a problem of machine translation from a natural language, English, to a formal control language.

The remainder of this paper is organized as follows. We describe two lines of research: first, learning to transform natural language route instructions to execution system inputs in an unfamiliar map, and second, learning a joint model of unfamiliar natural language and world percepts describing object attributes such as color and shape. We provide a brief overview of some related work in natural language grounding, instruction-following, vision, and learning, and conclude with a discussion of challenges encountered, expected future work, and the placement of teachable robotics in the broader context of HRI.

II. Learning to Follow Route Instructions

We explore the question of interpreting, or *grounding,* natural language commands so they can be executed by a robot, specifically in the context of following route instructions through a map [20]. Language grounding is treated as a question of semantic interpretation–that is, the extraction of a semantically meaningful representation of goals and world state from human-provided instruction text. This approach encompasses two key components: First, parsing natural

All authors are with the University of Washington, Department of Computer Science & Engineering, Seattle, WA 98195.

language into a formal representation capable of representing a robot and its operation in an environment; and, second, mapping the formal representation to actions and perceptions in the real world.

A. Goal

The key goal of this work is to learn grounding relations from data (rather than defining a mapping of natural language to actions), and to execute them against a previously unseen world model (in this case, a map of an environment). A statistical machine translation (SMT) approach is used to train a *parser,* which learns to parse natural language to expressive formal λ-calculus representations. The system learns the parser based on examples of English commands annotated by experts with the corresponding RCL expressions.

While λ-calculus and other formal languages have been used to represent complex robot control systems [6], [13], learning *from human interaction* requires this additional parser learning step, as non-expert end users are unlikely to be willing to write formal control language structures. Our goal in this work is a system that learns, from a reasonable number of examples, to parse NL commands into structures that capture the kind of procedural statements used by human instruction givers, such as counts, looping, and conditionals. Fig. 2 gives an overview of the approach.

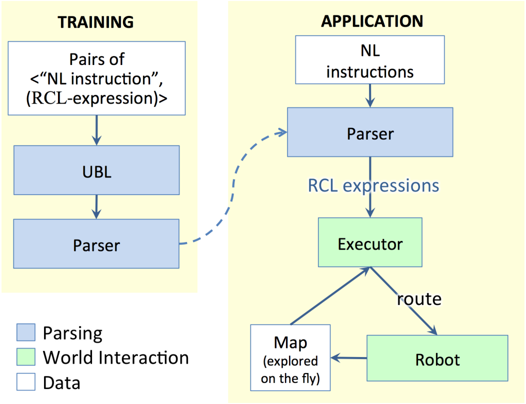


Fig. 2: The high-level architecture of the end-to-end system. During training, An English-to-RCL parser is learned. In testing, the learned parser maps NL instructions to an RCL program that is executed in simulation.

B. Approach

In practice, human route instructions include complex language, and robots must be able to execute these without being given a fully specified world model. We parse these complex instructions into correct, robot-executable commands for these complex NL instructions. Parses are expressed into a LISP-like control language called Robot Control Language, or RCL. [20].

The components of the system are as follows. First, paired corpora of natural language sentences and RCL commands were gathered. Approximately 25 non-experts gave route instructions describing randomly generated paths through a map, providing the natural language training data. Those sentences were annotated by experts (see Fig. 3 for an example). These corpora were given as input to the Unification-Based Learner, or UBL [14], [15], and a combined syntactic and semantic model was trained.

"Go left to the end of the hall."

```
(do-sequentially
  (turn-left current-loc)
  (do-until
    (or
      (not (exists forward-loc))
      (room forward-loc))
    (move-to forward-loc)))
```

"Go to the third junction and take a right."

```
(do-sequentially
  (do-n-times 3
    (do-sequentially
      (move-to forward-loc)
      (do-until
        (junction current-loc)
        (move-to forward-loc))))
  (turn-right current-loc))
```

Fig. 3: Examples of an English sentence and its associated RCL program. While short, this example demonstrates a few of the more complex concepts intrinsic to even simple natural language.

An important benefit of this approach is that the nature of our formal representation enables a robot to interpret route instructions online while moving through a previously unknown environment. The actual contextualization of the language into a world model happens as the map is discovered. The resulting system can represent control structures such as 'while,' higher-order concepts such as 'n$^{\text{th}}$,' and set operations, as well as following directions through unknown maps.

C. Parser Learning

RCL is a formal language defined by a grammar. To parse NL instructions into that language, we use an extended version of the Unification-Based Learner, UBL [14]. The grammatical formalism is *combinatory categorial grammars*, or CCGs [25], a type of phrase structure grammar. UBL creates a parser by inducing a CCG from a set of training examples (described above). CCGs model both the syntax (language constructs, such as NP for noun phrase) and the semantics (expressions in λ-calculus) of a sentence. The resulting RCL statement can be passed to a robot control system for execution.

Importantly, UBL can *learn* a parser solely from training data of the form $\{(x,z)\}$, where x is a natural-language sentence and z is a corresponding semantic-language sentence. In a nutshell, UBL learns a model for $p(z,y \mid x;\theta)$, where θ parameterizes the learned grammar G and y is a derivation in G. UBL uses a log-linear model: $p(z,y \mid x;\theta) \propto e^{\theta\cdot\phi(x,y,z)}$. UBL first generates a set of possibly useful lexical items, then alternates between increasing the size of this *lexicon* and estimating the parameters of G via gradient descent optimization (see [14] for more details).

D. Results and Discussion

For testing, a large set of routes through a previously unseen map was generated. These routes were divided into short (1 sentence) and long (an average of 5 sentences). English descriptions of those routes were generated by permuting known English phrases to describe elements of movement through the map, and those route instructions were parsed using the language model. Parses were tested by executing the resulting RCL programs in simulation. Table

I summarizes the percentage of path/NL pairs for which a simulated robot reached the destination successfully, on the desired route, by following the generated RCL program.

data set	success (short)	success (long)
enriched	92.4%	62.5%

Table I: Testing the end-to-end system on 1,000 short and 200 more complex sets of route instruction.

Execution was successful in 924/1000 of the short paths, and 125/200 of the complex paths. It is unsurprising that longer paths are more likely to fail than shorter ones: our simulated control system does not attempt any local error recovery if it encounters an unexpected situation, so a single poorly-parsed phrase in a route instruction means the robot cannot reach the destination.

Lessons and Challenges: In general, we believe this approach uses an effective approach to grounding natural language route instructions into a formal control language. Nonetheless, in the course of working with this data and system, a number of challenges were encountered. While the effort involved in expert annotation of natural language is far less than that of writing control system for the tasks, it still requires experts to be involved in the teaching process. Additionally, instructions through a map–and possibly instructions in general–tend to take a sequential form, in which individual instructions can be parsed separately with little loss of generality; however, this trait also makes local error recovery crucial, as shown by the results of our longer trials in Table I.

Lastly, we raise two points which we will touch on in the next section. Even a state of the art parser learning system does best with large amounts of training data, and gathering natural language from human subjects is extremely time-consuming. In our case, this resulted in a working set small enough to require artificial amplification (as described above). Furthermore, the language being gathered is still tied to our pre-defined grammar–the robot cannot be taught tasks which cannot be expressed using existing RCL symbols.

III. Learning Novel Object Attributes

The instruction-following framework described in the previous section has a number of desirable characteristics. The underlying parser-learning infrastructure is not particularly domain-dependent, and the resulting system generates correct RCL programs from NL in a reasonable number of test cases, although work remains. Nonetheless, it has some significant simplifications. One concern is that the set of people providing instructions was quite small, resulting in a need for artificial amplification in training and test cases [19], [20]. For this reason, in subsequent work, we explore gathering language via crowdsourcing.

A second, subtler concern is that, while previously-unseen English terms can be learned, the space of possible formal RCL programs that can be generated is limited by the set of predefined tokens appearing in the expert-provided RCL grammar. For many tasks, robots that can learn from natural language will need not only to learn how new natural language connects to existing concepts, but also to learn novel *concepts* in the underlying world model.

To address these concerns and to demonstrate the relative domain independence of our approach, we consider a second experiment: learning novel object attributes from a joint language and visual classification model [18].

A. *Goal*

Our goal in this work is to extend the framework described in Sec. II to incorporate visual percepts in order to learn about completely new color and shape attributes. This requires taking full advantage of physically grounded sensors and actuators in order to learn about objects in the environment. To do this, a robot must jointly reason about the data encountered, for example language and vision, and automatically induce rich associations. We target object attributes that might be defined both linguistically and visually; for example, “These are yellow cylinders,” uttered about a physical workspace that contains a number of objects that vary in shape and color. We assume that a robot can understand sentences like this if it can solve the associated *object selection task*–correctly identifying objects by characteristics encountered during a natural language teaching session.

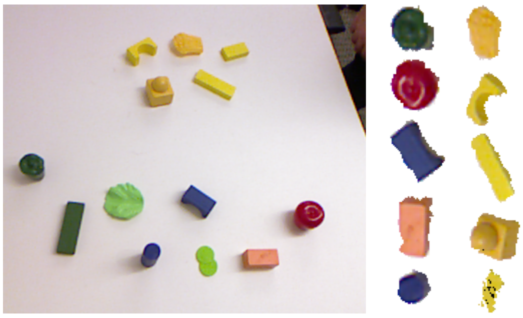


Fig. 4: An example of an RGB-D object identification scene. Columns on the right show examples of segmented objects. The far right column shows objects identified by the sentence “These are various types of yellow colored objects”; the center column shows non-selected objects. The correct associated RCL interpretation is λx(obj-color x, color-yellow).

There are a number of subproblems implicit in this goal. First, the robot must realize that words such as “yellow” and “square” refer to object attributes (unlike other unfamiliar words encountered during training, such as “thing” or “nearby”). It must also *ground* these words by mapping them to components of a perceptual system that enable it to identify the specific physical objects the language is referring to, even in cases where the attributes are entirely novel. Given the sentence “These are yellow” and the visual field shown in Fig. 4, the system must identify the cluster of blocks near the top of the image.

B. Approach

In order to learn the meanings of new words and associate them with new concepts, several components must be trained: visual classifiers that identify object properties, the meaning of individual words that may indicate these classifiers, and a model of compositional semantics that can be used to analyze sentences. We learn these components jointly, building on existing work on visual attribute classification [3] and the parser learning system described above.

In the experiment described in Sec. II, training data had two components, a natural language sentence x and an RCL expression y that formally captures the semantics of x. We now add additional terms: a set of scene objects O detected by a sensor, and the subset $G \subseteq O$ of objects described by x. We also require a set of *visual attribute classifiers* C, where each classifier $c \in C$ defines a distribution $P(c = true \mid o \in O)$ of the classifier returning true for each object o in the scene–for example, there would be a unique classifier c for each possible color or shape an object can have. We can then use logistic regression to train classifiers on color and shape features extracted from object segments detected with a Kinect depth camera.

Our key challenge is to learn to create new classifiers, associated with new NL words, which describe attributes not previously seen. We take a simple, exhaustive approach by creating a set of new classifiers which are initialized to uniform distributions. Each classifier is additionally paired with a new logical constant in RCL. Finally, a new parse rule is created by pairing each previously unknown word in a sentence with each new and existing classifier constant, with a very low initial likelihood of being used in parsing. This approach learns to jointly reestimate the parameters of both the new classifiers and the expanded parsing model. At the same time, the new classifiers are trained on each new object that is described by the word.

C. Experiments

We began by collecting a dataset of natural language sentences describing a variety of scenes. In order to avoid the problems described above of insufficient human data, we used Amazon's Mechanical Turk, a system which allows web-based distribution of small tasks to a group of workers in exchange for small payments [12]. Workers were asked to look at a scene and provide English descriptions of objects being gestured to (Fig. 5 shows an example). In this way, we gathered 1,003 natural language descriptions of objects appearing in 142 visual scenes.

Training is conducted in two phases. We conduct an initialization or 'bootstrapping' phase, in which we construct initial, limited language and perceptual models. We make use of a small supervised data set containing language and scenes, but in which we additionally label the latent logical form z and classifier outputs—for instance, the English word "yellow", the RCL constant `color-yellow`, and the associated classifier $c_{\texttt{color-yellow}}$. We subdivide our data such that this initialization does not contain words and attributes which will appear in training.

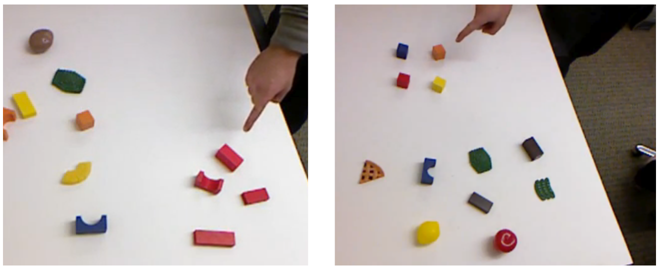
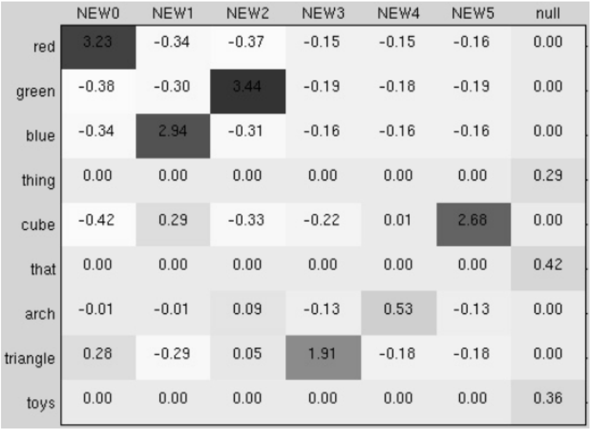


Fig. 5: Example scenes presented on Mechanical Turk. Left: A scene that elicited descriptions such as "here are some red things" and "these are various types of red colored objects", both of which would be labeled with the meaning $\lambda x.$`(objcolor` x`, red)`. Right: A scene associated with the sentence/meaning pair "this toy is an orange cube", $\lambda x.$`(objcolor` x`, orange)` $\wedge$ `(objshape` x`, cube)`.

We then jointly train the classifier model and language model on pairs of scene/language pairs. If this approach is successful, natural language words and phrases describing a new attribute will be consistently parsed to one of the new classifiers, which in turn should have good predictive power for identifying objects which have that attribute; words which do not denote an attribute or other RCL symbol should consistently parse to null. Fig. 6 shows an example of the parse likelihood of *lexemes*, or word-to-symbol mappings, after a typical run.

D. Results

We test for understanding via the object set selection task described above, using an 80/20 split of novel sentences for training/testing. The split of attributes into those used for bootstrapping and those used for training and testing is randomized over the span of 20 trials, and the order of sentences within each class is randomized. The robot is given a novel sentence and a scene with no objects marked, and precision, recall, and F1 score (a joint measure of



	NEW0	NEW1	NEW2	NEW3	NEW4	NEW5	null
red	3.23	-0.34	-0.37	-0.15	-0.15	-0.16	0.00
green	-0.38	-0.30	3.44	-0.19	-0.18	-0.19	0.00
blue	-0.34	2.94	-0.31	-0.16	-0.16	-0.16	0.00
thing	0.00	0.00	0.00	0.00	0.00	0.00	0.29
cube	-0.42	0.29	-0.33	-0.22	0.01	2.68	0.00
that	0.00	0.00	0.00	0.00	0.00	0.00	0.42
arch	-0.01	-0.01	0.09	-0.13	0.53	-0.13	0.00
triangle	0.28	-0.29	0.05	1.91	-0.18	-0.18	0.00
toys	0.00	0.00	0.00	0.00	0.00	0.00	0.36

Fig. 6: Lexeme weights at the end of a training run. The x-axis shows new natural language tokens discovered in training, and associated classifier symbols are given on the y-axis. In this example, each of three new colors and shapes has a single classifier that is strongly preferred, while the semantically empty tokens 'thing', 'that', and 'toys' have positive weight only in the lexeme mapping to null.

accuracy) are recorded for whether the objects described by the language are correctly identified.

Averaged across our trials, we obtain precision=0.81, recall=0.72, and F1=0.76 on the set selection task. This suggests that our integrated approach allows for reasonably effective learning with no explicit labeling of logical meaning representations or attribute classifier outputs. We also test the same data sets with the vision component removed, and with the language model replaced by simple keyword matching, to determine whether joint learning is necessary for our data. As expected, in both cases the system's performance degrades sharply. Results are summarized in Table II.

Approach	Precision	Recall	F1
Language only	0.52	0.09	0.14
Vision+keywords	0.92	0.41	0.55
Joint learning	0.81	0.72	0.76

Table II: A summary of the precision, recall, and F1 score for learning jointly versus ablations in which each component is reduced or removed. The vision-only system has good precision, because it generally identifies only a small number of objects.

Lessons and Challenges: We have presented a brief overview of a system that is able to successfully learn to generate formal expressions describing world characteristics from natural language paired with sensor inputs, using initialization data that does not contain the attributes we test against. Using Mechanical Turk for data collection was successful, and could readily lend itself to other 'teachable' problems–that is, those in which we wish to learn from non-expert data.

The learned model ise initialized with annotated data, and then trained on data that does not include the annotation. Given the cost of expert annotation, this is a definite step towards learning from interaction. Nonetheless, this initialization still requires some expert annotation, which is expensive and which means that the system cannot simply be deployed in a completely new domain with naïve users.

Perhaps more serious, the data efficiency of learning could be improved (as many positive and negative examples currently are required to train classifiers). This suggests that learning world models in a targeted, efficient way will be an important component in effective learning from real-time interaction.

IV. Related Work

Human-robot interaction is by nature a broad, cross-disciplinary problem, especially in the context of learning through interaction. Previous work in natural language grounding, human learning, formal representations, and vision, among others, are relevant. We highlight examples of relevant papers in this section.

With the advent of low-cost sensor platforms and strong language learning models, there has been significant work recently on grounded learning in the robotics and vision communities. Roy developed a series of techniques for grounding words in visual scenes [21], [23], [10]. In computer vision, language grounding often relates to detecting objects and attributes in visual information (e.g., see [1]), although these approaches primarily focus on isolated word meaning.

Interaction that takes perception into account relies heavily on vision. Object recognition has a clear place in teachable robotics, particularly in the context of building up a world model; current state-of-the-art systems [8], [27] are based on local image descriptors, for example SIFT over images [17] and Spin Images over 3D point clouds [11].

When trying to learn about new characteristics of objects in the world, visual attributes provide a rich source of information, and have become a popular topic in the computer vision community [7], [22]. Recent work on kernel descriptors [2] shows that these hand-designed features are equivalent to a type of match kernel that performs similarly to sparse coding [27], [28] and deep networks [16] on many object recognition benchmarks [2].

Approaches that learn probabilistic language models from natural language input [19], [5], especially those that include a visual component [26], [18], are closest to being directly usable in teachable robotics. [24] created a system that also learns to parse navigation instructions, but limited their formal language to a set of predefined parses.

Logic-based control systems have been used successfully in robotics [4], [9], [13], providing a framework for mapping language to robot control. In Dzifcak et al [6], it is demonstrated that NL commands can be mapped to robot controllers modeled as λ-calculus expressions, albeit with a manually constructed parser to map from NL commands to λ-calculus.

V. Conclusions

We draw several conclusions from these experiments. First, it is possible to use weakly supervised learning to learn a model of language and world-state able to handle complex natural language commands for robot instruction. This approach supports the learning of complex structures and entirely novel concepts. We find these results extremely encouraging with respect to the goal of learning to interpret rich NL instructions based on interaction with non-experts.

Increasingly available and capable robotics make pervasive, low-cost robots and robotic systems seem not only possible, but likely, in the near future. Successful human-robot interaction in this context will depend on the ability of those systems to learn from natural, intuitive interactions with these users—what we describe here as *teachable robotics*. The projects discussed highlight possible approaches, challenges, and avenues for further research; we believe that, when explored, this space will yield increasingly capable, effective pervasive robots.

References

[1] K. Barnard, P. Duygulu, D. Forsyth, N. De Freitas, D. Blei, and M. Jordan, "Matching words and pictures," *The Journal of Machine Learning Research*, vol. 3, pp. 1107–1135, 2003.

[2] L. Bo, X. Ren, and D. Fox, "Kernel descriptors for visual recognition," in *Neural Information Processing Systems (NIPS)*, 2010.

[3] ——, "Depth kernel descriptors for object recognition," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[4] C. Boutilier, R. Reiter, M. Soutchanski, and S. Thrun, "Decision-theoretic, high-level agent programming in the situation calculus," in *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2000.

[5] D. L. Chen and R. J. Mooney, "Learning to interpret natural language navigation instructions from observations," in *Proc. of the 25th AAAI Conference on Artificial Intelligence (AAAI-2011)*, August 2011, pp. 859–865.

[6] 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. of the 2009 IEEE Int'l Conf. on Robotics and Automation (ICRA '09)*, Kobe, Japan, May 2009.

[7] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[8] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE PAMI*, vol. 32, no. 9, pp. 1627–1645, 2009.

[9] A. Ferrein and G. Lakemeyer, "Logic-based robot control in highly dynamic domains," *Robotics and Autonomous Systems*, vol. 56, no. 11, 2008.

[10] P. Gorniak and D. Roy, "Understanding complex visually referring utterances," in *Proc. of the HLT-NAACL 2003 Workshop on Learning Word Meaning from Non-Linguistic Data*, 2003.

[11] A. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, 1999.

[12] A. Kittur, E. H. Chi, and B. Suh, "Crowdsourcing user studies with mechanical turk," in *Proc. of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, ser. CHI '08. ACM, 2008.

[13] H. Kress-Gazit, G. Fainekos, and G. Pappas, "Translating structured english to robot controllers," *Advanced Robotics*, vol. 22, no. 12, 2008.

[14] T. Kwiatkowski, L. Zettlemoyer, S. Goldwater, and M. Steedman, "Inducing probabilistic CCG grammars from logical form with higher-order unification," in *Proc. of the Conference on Empirical Methods in Natural Language Processing*, 2010.

[15] ——, "Lexical generalization in CCG grammar induction for semantic parsing," in *Proc. of the Conference on Empirical Methods in Natural Language Processing*, 2011.

[16] H. Lee, R. Grosse, R. Ranganath, and A. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. of the International Conference on Machine Learning (ICML)*, 2009.

[17] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision (IJCV)*, vol. 60, pp. 91–110, 2004.

[18] C. Matuszek, N. FitzGerald, L. Bo, L. Zettlemoyer, and D. Fox, "A Joint Model of Language and Perception for Grounded Attribute Learning," in *Proc. of the International Conference on Machine Learning (ICML)*, Edinburgh, Scotland, June 2012.

[19] C. Matuszek, D. Fox, and K. Koscher, "Following directions using statistical machine translation," in *HRI 2010: Proc. of the 5th Int'l Conf. on Human-Robot Interaction*. ACM Press, 2010.

[20] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox, "Learning to parse natural language commands to a robot control system," in *Proc. of the 13th International Symposium on Experimental Robotics (ISER)*, June 2012.

[21] N. Mavridis and D. Roy, "Grounded situation models for robots: Where words and percepts meet," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.

[22] D. Parikh and K. Grauman, "Relative attributes," in *International Conference on Computer Vision*, 2011.

[23] H. Reckman, J. Orkin, and D. Roy, "Learning meanings of words and constructions, grounded in a virtual game," in *Proc. of the 10th Conference on Natural Language Processing (KONVENS)*, 2010.

[24] N. Shimizu and A. Haas, "Learning to follow navigational route instructions," in *Int'l Joint Conf. on Artificial Intelligence (IJCAI)*, 2009.

[25] M. Steedman, *The Syntactic Process*. MIT Press, 2000.

[26] 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. of the National Conference on Artificial Intelligence (AAAI)*, August 2011.

[27] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[28] K. Yu and T. Zhang, "Improved local coordinate coding using local tangents," in *Proc. of the International Conference on Machine Learning (ICML)*, 2010.