Toward Interactive Grounded Language Acquisition

Thomas Kollar, Jayant Krishnamurthy, Grant Strimel
Computer Science Department, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213

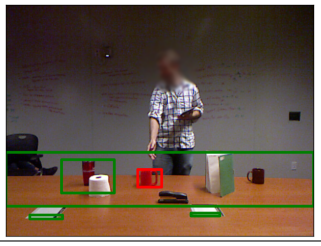
***Abstract*—This paper addresses the problem of enabling robots to interactively learn visual and spatial models from multi-modal interactions involving speech, gesture and images. Our approach, called Logical Semantics with Perception (LSP), provides a natural and intuitive interface by significantly reducing the amount of supervision that a human is required to provide. This paper demonstrates LSP in an interactive setting. Given speech and gesture input, LSP is able to learn object and relation classifiers for objects like mugs and relations like left and right. We extend LSP to generate complex natural language descriptions of selected objects using adjectives, nouns and relations, such as "the orange mug to the right of the green book." Furthermore, we extend LSP to incorporate determiners (e.g., "the") into its training procedure, enabling the model to generate acceptable relational language 20% more often than the unaugmented model.**

I. Introduction

As robots move out of the lab and into the real world, it is critical to develop methods for human users to flexibly and intuitively interact with them. Multi-modal interaction is a compelling solution since the operator can flexibly specify complex requirements with speech, gesture and vision. This paper addresses the challenge of teaching robots about objects and relations from natural language and gesture input.

We address this problem by developing a system that is able to perform grounded language acquisition, learning to map from natural language expressions to their referents in the world, called *groundings*. Two critical capabilities for this setting are (1) *interactive learning*, the ability to learn the real-world referents of words directly from interactions with people, and (2) *language generation*, the ability to construct novel natural language expressions describing real-world objects. For example, imagine that a person points at an object and says, "This mug is on the table." From this interaction, an autonomous robot should learn (1) which objects in the environment can be referred to by "mug" and "table" and (2) the spatial relation referred to by "on." Furthermore, the robot should be able to describe objects in new environments using novel combinations of these words.

The interactive setting is challenging for grounded language acquisition since only a portion of the correspondence between language and the environment can be inferred from real-world interaction. Consider the example of "this mug is on the table" with an accompanying gesture indicating the real-world referent of the complete phrase. In this interaction the robot does not observe the real-world referent of "table," since the user only gestured toward the mug, which means that the robot cannot be certain about which objects participate in the spatial relation given by "on." Furthermore, if there is only a single mug in the environment, this interaction provides no negative information about which objects are not "on the table." To enable interactive grounded language acquisition, the robot must be able to learn from such partially observed language/environment mappings.



Generated Natural Language:
"the orange mug is to the right of the green book"

Fig. 1: Example natural language output from our model.

This paper builds off recent work on Logical Semantics with Perception (LSP) [14], a weakly-supervised yet expressive approach to grounded language acquisition. LSP is suitable for the interactive setting because it is trained directly from language/real-world referent pairs. This supervision requirement contrasts with other approaches that use annotated semantic parses or a fully-observed correspondence between language and the world [12, 27, 21]. Furthermore, LSP is more expressive than these approaches, as it learns relations between sets of objects in addition to categories. These properties make LSP an ideal starting point for interactive grounded language acquisition.

The LSP model from [14] learns to map from language to objects and relations in the environment. This paper extends LSP to the interactive setting, by considering language generation and introducing novel constraints on learning that improve performance. The primary contributions of this paper are:

- **Language Generation**. An approach for generating language referring to objects in the environment with LSP. Our approach is capable of generating complex, relational descriptions such as "the orange mug to the right of the green book." (Figure 1)
- **Determiners**. We present novel semantic constraints on determiners and incorporate them into model training. These constraints are particularly important for learning relations: we find that they improve language generation performance by 20% on relational language.
- **Demonstration/Experiments**. We provide results with an interactive system, showing that LSP correctly maps language to its real-world referent and is able to generate correct relational object descriptions. Our system enables users to point at objects and describe them in natural language (Figure 4); these interactions are captured using

an RGB-D camera and gesture recognition and are used to train LSP.

II. Related Work

Beginning with SHRDLU [29], many systems have modeled the compositional structure of language to understand a natural language command [23, 25, 19, 7]. Typically, these systems model either compositional semantics or perception, instead of jointly modeling both. There has also been work on embodied vision to learn objects and their properties, though language was not a major component [1].

Semantic parsing is the area of compositional semantics most related to this work. This problem has been studied under various supervision assumptions, the most typical of which assumes observed logical forms [30, 31, 10, 17]. More relevant to this work are weakly-supervised formulations [5, 18, 15], which observe functions of the output semantic parse. However, these works all rely on manually constructed knowledge representations. Similarly, there is work on grounded language acquisition using a formal representation of the environment [11, 3, 24, 20, 4]. Similar approaches have been used to translate natural language into robot commands [16, 13, 7, 2], where the components of the meaning representation are primitive robot behaviors. We note that [2] also models determiners, but does not consider the interaction between determiners and perception.

The approach taken in this paper is to jointly learn a model of perception with a model of language acquisition. Our work extends Logical Semantics with Perception (LSP) [14], an expressive and weakly-supervised model that jointly learns to semantically parse text and ground both categorial and relational language in an environment. LSP is more flexible than work on robot direction following [24, 12, 27], which does not represent parse structure ambiguity. LSP also extends work on attribute learning [21] by learning two-place relations in addition to one-place categories. Previous work on LSP [14] did not consider constraints on determiners or a method for generating natural language referring expressions.

III. Logical Semantics with Perception

This section reviews Logical Semantics with Perception (LSP), a model for grounded language acquisition introduced in [14]. LSP accepts as input a natural language statement and an image and outputs the objects in the image denoted by the statement. The LSP model has three components: perception, parsing and evaluation (see Figure 2). The perception component constructs logical knowledge bases from low-level feature-based representations of images. The parsing component semantically parses natural language into lambda calculus queries against the constructed knowledge base. Finally, the evaluation component deterministically executes this query against the knowledge base to produce LSP's output. The output of LSP can be either a *denotation* or a *grounding*. A denotation is the set of image segments referred to by the complete statement, while a grounding is a set of image segment tuples representing the referents of each noun phrase in the statement. Figure 2(c) shows the distinction between these outputs. The LSP model can be trained using only language/denotation pairs, which only partially specify the mapping between language and the environment, yet are a natural form of supervision in the interactive setting. A training example for the environment in Figure 2(a) could be, "the red mug on the table" along with the denotation, which would consist of image segment 3.

Mathematically, LSP is a linear model f that predicts a denotation γ for a natural language statement z in an image d. The model factorizes according to the structure sketched above, using several latent variables:

$$f(\gamma, \Gamma, \ell, t, z, d; \theta) = f_{prs}(\ell, t, z; \theta_{prs}) + f_{gnd}(\Gamma, d; \theta_{gnd}) + f_{eval}(\gamma, \Gamma, \ell) \quad (1)$$

The LSP model assumes access to a set of category ($c \in C$) and relation predicates ($r \in R$), which are automatically derived from the training data. The f_{gnd} function perceives the world, taking an image d and producing a logical knowledge base Γ using the given vocabulary of predicates and parameters θ_{gnd}. The f_{prs} function represents a semantic parser. This component produces a logical form ℓ and a syntactic parse tree t for the natural language statement z, given parameters θ_{prs}. Finally, f_{eval} represents the deterministic evaluation component, which evaluates the logical form ℓ on the knowledge base Γ to produce a denotation γ. Each of these components is sketched in more detail in the following sections. An illustration can be seen in Figure 2.

A. Perception Function

The perception module takes an image as input and produces a logical knowledge base representing the agent's beliefs. The perception function runs classifiers on each image segment to determine category membership (e.g., $\texttt{mug}(x)$) and on pairs of image segments to determine relation membership (e.g., $\texttt{left-rel}(x, y)$). Let $\gamma^c \in \Gamma$ denote the set of image segments which are elements of a category predicate c; similarly, let $\gamma^r \in \Gamma$ denote the pairs of image segments which are elements of the relation predicate r. Given these sets, the score of a logical knowledge base Γ factors into per-relation and per-category scores h:

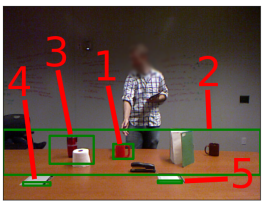
$$f_{gnd}(\Gamma, d; \theta_{gnd}) = \sum_{c \in C} h(\gamma^c, d; \theta_{gnd}) + \sum_{r \in R} h(\gamma^r, d; \theta_{gnd})$$

The per-predicate scores are in turn given by a sum of per-element classification scores:

$$h(\gamma^c, d; \theta^c_{gnd}) = \sum_{e \in E_d} \gamma^c(e)(\theta_{gnd})^T \phi_{cat}(e)$$

$$h(\gamma^r, d; \theta^r_{gnd}) = \sum_{(e_1, e_2) \in E_d} \gamma^r(e_1, e_2)(\theta_{gnd})^T \phi_{rel}(e_1, e_2)$$

The input to each category classifier is an image segment e, described by a feature vector $\phi_{cat}(e)$, and the output is either true or false. Similarly, the the input to each relation classifier is a pair of image segments (e_1, e_2) described by a feature vector $\phi_{rel}(e_1, e_2)$, and the output is either true or false. In the above equations, denotations γ are treated as indicator



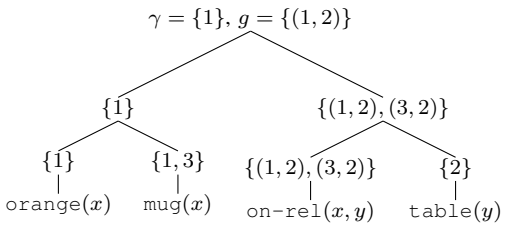
$\Gamma = \{\texttt{mug}(1), \texttt{mug}(3), \texttt{orange}(1), \texttt{table}(2), \texttt{on-rel}(1,2), \texttt{on-rel}(3,2), \texttt{left-rel}(3,1)\}$

(a) Perception

z="orange mug on the table"

$\ell = \lambda x.\exists y.\texttt{orange}(x) \wedge \texttt{mug}(x) \wedge \texttt{on-rel}(x,y) \wedge \texttt{table}(y)$

(b) Parsing



(c) Evaluation

Fig. 2: Illustration of LSP's three model components. Perception (a) takes an environment (image) containing a set of image segments and applies a set of per-predicate perceptual classifiers to produce a logical knowledge base Γ representing the agent's beliefs. Parsing (b) uses a semantic parser to map the user's language z to a lambda calculus statement ℓ. Evaluation (c) evaluates this lambda calculus statement ℓ on the knowledge base Γ to produce a denotation γ and grounding g.

functions for the sets they denote, i.e., $\gamma(e) = 1$ if e is in the set. Both kinds of perceptual classifiers are linear classifiers, parametrized by a single feature vector per predicate. The set of all of these feature vectors is represented by θ_{gnd}. Each detected predicate becomes a ground predicate instance in the output knowledge base Γ (see Figure 2(a)).

B. Semantic Parsing

LSP uses a Combinatory Categorial Grammar (CCG) to parse natural language statements into logical forms. Logical forms are lambda calculus functions that select a subset of image segments from the environment. A CCG is defined by a lexicon, which assigns meanings to individual words:

$$\begin{aligned} \text{orange} &:= N/N : \lambda f.\lambda x.\texttt{orange}(x) \wedge f(x) \\ \text{mug} &:= N : \lambda x.\texttt{mug}(x) \\ \text{on} &:= (N\backslash N)/N : \lambda f.\lambda g.\lambda x. \\ &\quad \exists y.f(y) \wedge g(x) \wedge \texttt{on-rel}(x,y) \end{aligned}$$

The meaning of a word in CCG is a *function* with both syntactic and semantic components. The syntactic component (e.g., $N, N/N$) represents how the word can be combined with adjacent words during parsing. For example, the "orange" entry has syntax N/N, which means that it takes a noun as its argument on the right. The semantic component maps each word to a lambda calculus expression. Parsing uses a set of rules (e.g., function application) to combine these expressions and produce a logical form. For example, using the above lexicon, the function for "orange" can be applied to the function for "mug," producing the parse $\lambda x.\texttt{mug}(x) \wedge \texttt{orange}(x)$ for $z =$"orange mug." A complete parse of "orange mug on the table" is shown in Figure 2(b).

In CCG, a single sentence has many parses. To disambiguate between these options, LSP uses features ϕ_{prs} to train a linear model over semantic parses ℓ, t:

$$f_{prs}(\ell, t, z; \theta_{prs}) = \theta_{prs}^T \phi_{prs}(\ell, t, z)$$

For additional details on f_{prs}, the reader should consult [14].

C. Evaluation

The evaluation module combines a logical form ℓ and a logical knowledge base Γ to produce a denotation γ and a grounding g. For example, consider the logical form in Figure 2(b). Evaluation first looks up the individual predicate instances using Γ (leaves in Figure 2(c)). The results are then combined using the following recurrences:

- If $\ell = \lambda x.\ell_1(x) \wedge \ell_2(x)$, then $\gamma(e) = 1$ iff $\gamma_1(e) = 1 \wedge \gamma_2(e) = 1$.
- If $\ell = \lambda x.\exists y.\ell_1(x,y)$, then $\gamma(e_1) = 1$ iff $\exists e_2.\gamma_1(e_1, e_2) = 1$.

This evaluation of ℓ on Γ returns a denotation, which is the set of image segments for which the logical form is true.

D. Parameter Estimation

LSP can be trained either using full or weak supervision. Full supervision involves annotating correct semantic parses and logical knowledge bases for a collection of natural language statements and images. Fully supervised training is laborious, unnatural, and inappropriate for an interactive setting. Instead, we use a weakly supervised training procedure that estimates parameters directly from language/denotation pairs, which provide only a partial mapping between natural language and the environment. In Figure 2, the language would be "orange mug on the table" and the denotation would be $\gamma = \{1\}$. Such training is more appropriate for our interactive setting, as we can obtain language via speech recognition, and can obtain a denotation from the user's pointing gesture. Training jointly optimizes the parameters of the semantic parser (θ_{prs}) and of the perceptual classifiers (θ_{gnd}) to predict correct denotations for the training instances. The model is trained as a max-margin Markov network (M^3N) [26] using stochastic subgradient descent; we refer the reader to [14] for details.

IV. Language Generation

In an interactive setting, it is desirable for a robot to generate natural language descriptions of its visual scene. Ideally, given a denotation γ^* and an image d, a language model $p(z)$ would be incorporated into a generative model (with the same factorization as LSP) as follows:

$$\arg\max_{z,\Gamma,\ell,t} p(z) \times p(\gamma^*, \Gamma, \ell, t | z, d, \theta)$$

Unfortunately, learning such a generative model is intractable because it requires marginalizing over logical knowledge bases subject to constraints given by the training data.

Similarly, LSP's scores are not probabilities, so they cannot be directly combined with a language model. As a result, we propose to generate language by maximizing the likelihood of the generated text according to a language model, while using LSP to ensure that the text refers to the appropriate set of objects. Given a denotation γ^* and an image d, language generation aims to calculate:

$$\arg\max_z p(z) \quad (2)$$
$$\text{s.t.} \quad \arg\max_\gamma \left(\max_{\Gamma,\ell,t} f(\gamma, \Gamma, \ell, t, z, d; \theta) \right) = \gamma^*$$

f represents a trained LSP model with parameters θ. The model $p(z)$ is bigram language model that defines a probability distribution $p(z)$ over word sequences z as follows:

$$p(z) = p(z_1|\text{START}) \left(\prod_{i=2}^{n} p(z_i|z_{i-1}) \right) p(\text{END}|z_n)$$

The per-word distributions above are multinomial distributions over words, estimated from the training corpus.

We propose a search algorithm that approximately maximizes this objective. The aim of inference is, given a denotation (set of image segments) γ^*, to produce a natural language expression describing those segments. The first step of inference is to generate all adjective/noun pairs that could be referring expressions to γ^* (e.g., "orange mug."). This is done by running LSP on each candidate adjective/noun pair. If the highest-scoring denotation produced by LSP does not contain γ^*, then the adjective/noun pair is discarded. The second step of inference is to generate, for each image segment, the most likely adjective/noun phrase that uniquely refers to that image segment. This is done by running all phrases through LSP, maintaining the highest-scoring expression (according to the language model) that describes each image segment. In both steps, an article ("a" or "the") is optionally prepended to each phrase to improve the fluency of the text.

Next, the procedure generates all relation expressions such as "right" and "left" using the lexicon. All of the prepositions that can apply to these expressions are then prepended and appended to produce relation expressions such as "to the right of." These relation expressions are then combined with the highest-scoring expressions for each image segment (Step 2), producing phrases such as "to the right of the cup." LSP is then run on each of these phrases to identify whether γ^* is included in the first argument of their denotations.

Given the resulting natural language expressions for γ^* and expressions that can describe the relationship between γ^* and all of the other image segments, the search then scores all combinations of these expressions using the language model $p(z)$, identifying the single most probable way to describe γ^*. The final result of generation is a noun phrase, such as "the orange mug to the right of the green book." To generate more complicated natural language statements, the relation generation step can be performed recursively.

V. Determiners

Determiners, like "a" and "the," provide important cues that can help guide the interpretation of language. For example, if a user utters "the mug," we can infer that (1) the denotation of "the mug" contains exactly one image segment, and (2) there is a unique image segment which can be called "mug." The second inference is especially valuable in the interactive setting, as it allows the model to generate negative examples of mugs, which are often unavailable from user interactions. This section describes an extension to LSP that enables it to incorporate the effects of determiners into both parameter estimation and inference.

In previous work, determiners like "a" and "the" were assigned the lexicon entry $N/N : \lambda f.f$ [14]. This lexicon entry states that both words have no meaning: a semantic parser using this entry for "the" produces the same logical form ($\lambda x.\texttt{mug}(x)$) for both "the mug" and "mug." Here, we add presupposition [9] constraints to these words, which allow them to influence perception:

the $:= N/N : \lambda f.\lambda x.f(x)$ Presuppose: $\exists!y.f(y)$
a $:= N/N : \lambda f.g$
Presuppose: $\exists g.(\forall x.g(x) \Rightarrow f(x)) \wedge (\exists!y.g(y))$

The presupposition clauses behave like constraints on the result of perception: intuitively, if a human utters "the mug," but the robot believes there are two mugs in the scene, then the robot is incorrect and should revise its perception of the scene to agree with the human's utterance. The logical form for "the" captures this intuition, stating that the denotation of "the mug" is the same as the denotation of "mug," while its presupposition states the robot should believe the world contains exactly one mug ($\exists!$ denotes a unique existential quantifier). Similarly, we expect "a mug" to denote a single mug, although there may be multiple mugs in the image. The above formula for "a" therefore asserts that (1) the denotation for "a mug" contains at most one image segment (i.e., $\exists!y.g(y)$), but (2) that image segment can be any single image segment from the denotation of "mug" (i.e., $(g(x) \Rightarrow f(x))$). Note that there exist multiple possible functions g that satisfy this constraint, capturing our intuition that the word "a" represents an arbitrary choice of a referent from a set possible referents.

During parameter estimation, we use the presupposition clauses from the determiners as an additional source of supervision. If a phrase contains a determiner, LSP's perceptual classifiers are trained to satisfy the phrase's presupposition constraints. As an example, consider the phrase "mug on the table." Without determiner constraints, LSP training allows "table" to refer to an arbitrary set of objects in the image. With determiner constraints, LSP training encourages "table" toward an interpretation that refers to a single object. If our determiner constraints are true in the data set, then such training should improve our category and relation classifiers by creating negative examples of both "table" and "on." To use determiners as a source of supervision, the presupposition clause constraints are enforced during the subgradient computation when finding the best semantic parse and knowledge

base that explain a labeled denotation.

The presupposition constraints also increase the difficulty of inference since, during the evaluation step (Figure 2(c)), the robot must produce a logical knowledge base Γ that agrees with the constraints. In the unaugmented LSP, the denotation of a natural language statement can be computed efficiently due to the deterministic structure of the evaluation module f_{eval}. However, the determiner constraints allow the logical form to influence the logical knowledge base during the evaluation step. For example, if the knowledge base states that there are two mugs in the scene, but the natural language statement is "the mug," inference must revise the predicted knowledge base such that it contains only one mug. To address these issues, the presupposition constraints are incorporated into the integer linear program (ILP) inference algorithm given in [14]. This ILP explicitly encodes uncertainty over the logical knowledge base along with the sets of image segments produced by each stage of evaluation (as shown in Figure 2(c)). The presupposition constraints are encoded using hard constraints on these sets of image segments.

VI. Interactive Training

This section describes the interactive scene understanding system that we constructed for our experiments. Our system enables humans to interactively train LSP to recognize objects on a table using pointing gestures and speech. The system consists of an RGB-D camera and an Android tablet. The RGB-D camera is used to track people and segment the scene; the tablet is used for both its speech to text and text to speech capabilities. Low-level communication is handled by LCM [8] and ROS [22].

There are two modes for the system. The training mode enables a user to generate training examples for LSP, while the generation mode allows a user to obtain natural language descriptions of objects. In training mode, the user points at an object in the scene. The system detects the pointing gesture, then initiates the following dialog:

- Robot: *Please describe what you are pointing at.*
- Person: *There is a mug on the table.*
- Robot: *Training my model on the text, "There is a mug on the table."*

From this interaction, the system collects (1) an image and an accompanying set of segments, (2) the pointed-at segment, and (3) a natural language statement produced from automatic speech recognition. This data can be immediately fed into a continuously-training LSP model, or saved for offline analysis. In generation mode, a user can ask the system to describe an object by pointing. The pointed-at object is sent to LSP, which automatically generates a natural language description of the object in the context of the surrounding scene. The natural language generation mode of the system is shown in Figure 4.

A. Segmentation

We have implemented an approach which automatically segments objects from a scene; the stages of this approach are shown in Figure 3. A region growing segmentation algorithm uses the RGB-D image to compute neighbors (in depth) of a region, then adjacent regions are merged based on their color-averages to produce a segmentation. To train LSP, we retained only the image component of each resulting RGB-D segment by transforming it into an axis-aligned bounding box on the RGB component. The bounding boxes surrounding each object are displayed on a screen visible to the user. To reduce the number of false positives produced by segmentation, we filter out segments behind the user as well as bounding boxes that have an aspect ratio greater than 1:10 (e.g., a bounding box with aspect ratio 1:100 would be filtered). The basic segmentation algorithm is a part of the Point Cloud Library (PCL).

B. Pose Tracking

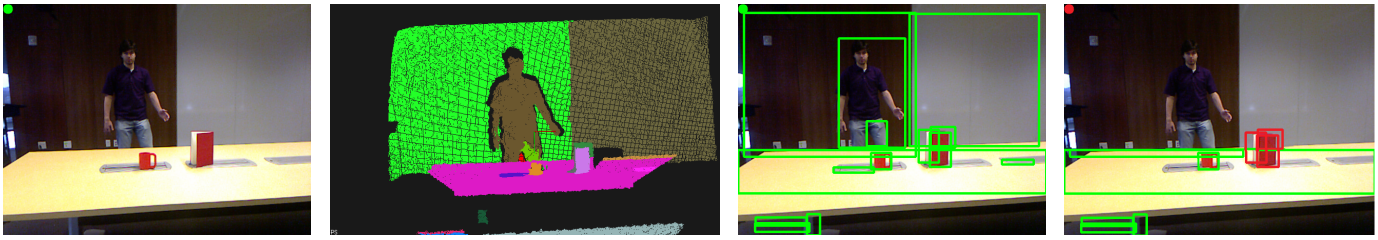
Our system recognizes gestures by tracking the user using the OpenNI human tracker. The user initiates tracking by standing in front of the tracker in a pose with their arms up. Once tracking has started, the user can point at objects with their right hand. These pointing gestures are recognized by extending the vector from the right elbow to the right hand outward. We compute bounding box intersections with this vector by performing ray tracing in three dimensions to determine where the person is pointing. If the elbow-hand ray intersects a segmentation bounding box for more than 15 frames, the system signals that the object is being pointed at.

VII. Results

We performed two experiments in our interactive setting to understand the performance impact of determiner constraints, and to evaluate language generation. The first experiment evaluates LSP+det's ability to identify objects from natural language descriptions, and the second experiment evaluates LSP+det's ability to describe objects in natural language.

We created a data set for our evaluation using our interactive system. Three users interacted with the system, gesturing at objects and describing them in spoken language. The positions of objects in the scene were varied several times during each user interaction. The images, segmentations, denotations, and speech recognition output were captured and stored for offline analysis. In order to perform a fair evaluation, we manually examined the data set and removed instances where the indicated object did not match the user's description. In all, we discarded 43% of the examples. 66% of the discarded examples were due to segmentation errors (e.g., the described object is not in the segmentation) and the remaining 34% due to transcription errors (e.g., "Green Book we need bread coffee cup" should be "green book beneath the red coffee cup"). Many of the segmentation errors were identified by users during the interaction, in which case users were instructed to say "nothing." The resulting data set contains 94 training examples, with 71 distinct natural language descriptions, split across three users. For both sets of experiments, we performed three-fold cross-validation, training on the examples generated by two users and testing on the held-out user.

We noticed that, during interactive training, users were overwhelmingly likely to generate language which uniquely



(a) RGB-D Image (b) Initial Segmentations (c) Axis-aligned Bounding Boxes (d) Filtered using user's depth

Fig. 3: The segmentation takes an (a) RGB-D image (b) performs segmentation (c) extracts axis-aligned bounding boxes and (d) filters the bounding boxes by depth and aspect ratio. In (d) the user is pointing at the red bounding boxes.

identified an object without requiring the system to interpret a relation. For example, the correct denotation for the phrase "the green book on the table" can be identified solely from "the green book," if the image only contains one green book. 50 of the examples contain relational language, of which only 7 require the relation to be interpreted. Users were not instructed to provide such interactions – they appear to be a natural property of the way people describe objects.

This property of the data set has two consequences. First, learning relations is more challenging because the examples do not provide negative relation instances. In the above example, we cannot determine what objects are *not* "on the table." The determiner constraints help LSP learn relations from these difficult training instances by further constraining the denotations of the relation's two arguments. Second, denotations are easy to predict without understanding relations. This property hides differences in relational understanding in the first evaluation (Section VII-B), since most denotations can be predicted using only the first noun phrase in a statement. However, regardless of whether or not relations are required to predict which object is being described, we would still like our model to learn relations (e.g., for language generation). Our language generation experiments (Section VII-C) reveal the improvements in relational understanding produced by incorporating determiner constraints.

A. Preprocessing and Features

This section describes the initialization of LSP with a set of category and relation predicates, the construction of the CCG lexicon, and the features used by the perceptual classifiers to detect concepts in the image.

The predicates and CCG lexicon are automatically generated from the training data, with no manual intervention. We apply the same lexicon induction heuristics as in [14]. This procedure allows the formal representation used by the robot to grow based on the language uttered by users. First, we part-of-speech tag and lemmatize the input language with the Stanford CoreNLP pipeline [28]. Next, we create a category predicate for each word that is tagged as a noun ("mug") or adjective ("orange"), and we create a relation predicate for words that are tagged as nouns ("right"), prepositions ("on"), or verbs ("is"). Predicates are named after a word's lemma, and relations are suffixed with `-rel`. Each time a predicate is created in this fashion, we also create a lexicon entry for the CCG parser that maps the word to the corresponding predicate, with an appropriate syntactic type for the POS tag. The lexicon also includes a few special entries for forms of "to be." To create the LSP+det model, we replace the lexicon entries for "the" and "a" as described in Section V. In total, the lexicon contains 293 entries, with 31 category and 13 relation predicates. On manual inspection, we found that 19 of the generated categories and 10 of the generated relations actually occurred in scenes; the other predicates were generated by minor speech recognition errors and ambiguity about whether a word should invoke a category or relation.

There are two feature sets for the perceptual classifiers: one for categories and one for relations. The category features include a Histogram of Oriented Gradients (HOG) [6], and an RGB color histogram. These features allow the model to represent both shape-based and color-based properties. The relation classifiers use a set of spatial features computed from pairs of bounding boxes. Examples of these features include bounding box overlap area, and directional features capturing the relative positions of two bounding boxes.

B. Denotation Evaluation

The first evaluation measures our model's ability to correctly predict an object being described by a user. In this evaluation, we give our system each of the held-out user's utterances, then calculate the most likely denotation for the utterance. We measure performance using accuracy, which is the fraction of examples for which the system predicts the same denotation indicated by the user. The goals of this evaluation are to determine (1) the impact on performance of incorporating determiner constraints, and (2) whether LSP's performance is competitive with other approaches. As noted above, the nature of users' utterances means that this evaluation is mostly a test of a model's ability to represent categorial language. Relational language is tested in the following evaluation.

Our first evaluation trains and tests on the complete natural language statements generated by each user. For this evaluation, we considered two baseline models. The first is a random baseline, which selects a random segment of the image as the described object; this baseline represents chance performance on this task. The second model is the original LSP model, without special processing of determiners. When making predictions with LSP and LSP+det, we selected the highest-scoring denotation containing exactly one object. Without this constraint, these models predict sets of objects (i.e., possibly more than one object), which is more expressive

	User 1	User 2	User 3
LSP+det	0.5	0.375	0.40
LSP	0.55	0.375	0.38
Random	0.17	0.15	0.15

TABLE I: Denotation prediction accuracy of LSP+det and the baselines. Predictions are computed using the entire natural language statement generated by each user.

	User 1	User 2	User 3	Overall (micro avg.)
LSP+det	0.60	0.38	0.5	0.48
LSP	0.55	0.38	0.40	0.42
SVM	0.45	0.41	0.48	0.45
Random	0.17	0.15	0.15	0.16

TABLE II: Denotation prediction accuracy of LSP+det and the baselines for categorial language. Predictions are computed using only the first noun phrase of each statement.

than necessary for our data set. The results of this evaluation are shown in Table I. Both LSP and LSP+det perform significantly better on this task than the random baseline. However, due to the categorial nature of this task, both LSP and LSP+det perform quite comparably.

To demonstrate that LSP is competitive with other approaches, we ran a categorial language experiment. We constructed a categorial language data set by extracting the first noun phrase from each example, discarding all following language. For constructions like "this is X," we extracted the first noun phrase of "X." The resulting data set contains the same number of language/denotation pairs as our original data. We then trained a Support Vector Machine (SVM) baseline using this data set. This baseline does not model the compositional structure of language, instead training a single SVM classifier per noun phrase to predict its denotation (e.g., there is an SVM for "green book" and an independent SVM for "book"), using the same features as the LSP models.

We compared the SVM baseline to LSP and LSP+det on the categorial language data set. The results of this evaluation are shown in Table II. In this experiment, the SVM model is trained using the categorial language examples, while LSP and LSP+det are trained on the complete natural language statements. Overall, we can see that all three models perform comparably, though LSP+det may have slightly better performance than LSP and the SVM baseline. These results suggest that LSP and LSP+det, when trained on complete natural language statements, learn categorial language as well as a fully supervised learner, even though they are solving a harder learning problem.

C. Language Generation

We additionally performed an experiment to measure the quality of the natural language object descriptions generated by LSP and LSP+det. For each example in each held-out fold of our data, we used LSP and LSP+det to generate a description of the indicated object in the scene. The generated descriptions have the form noun-relation-noun, with optional adjectives associated with each noun. For each resulting description, we manually evaluated whether each adjective, noun and relation phrase in the generated text accurately described the indicated object. Since adjectives were optional, we also counted the number of phrases where both nouns and the relation phrase were correct. Every generated description contains a relation phrase, so this experiment thoroughly evaluates each model's understanding of relations.

	Adjectives	Nouns	Relations	Nouns+Relations
LSP+det	0.68	0.77	0.59	**0.39**
LSP	0.58	0.77	0.39	**0.25**

TABLE III: Language generation accuracy of both LSP and LSP+det, measured using cross-validation across all three folds of our data set.

Table III shows the results of the language generation evaluation, and Figure 4 shows some example language generated by the two models. While both LSP and LSP+det perform comparably on nouns, LSP+det dramatically outperforms LSP on relations. This result indicates that LSP+det has learned a more accurate model of relations due to its use of determiner constraints. Furthermore, many of the mistakes made by LSP+det are quite reasonable, given its limited feature representation. For example, our visual features make it difficult to distinguish "beneath" from "behind"; such a mistake is manifested in Figure 4(c).

VIII. Conclusion

This paper presents LSP+det, a multi-modal model that is able to learn how to interpret and generate natural language expressions involving categories and relations from images and gestures. LSP+det extends existing work on LSP by incorporating determiners into its training procedure. We further introduce a procedure for generating complex natural language descriptions of objects using adjectives nouns, and relations, and show that LSP+det generates better natural language than the unaugmented LSP. These results indicate that better models of relations can be learned by taking advantage of subtle aspects of natural language like determiners. Furthermore, we present an interactive system that can be used to teach robots about objects and relations with no prior knowledge of the language or scene.

Acknowledgments

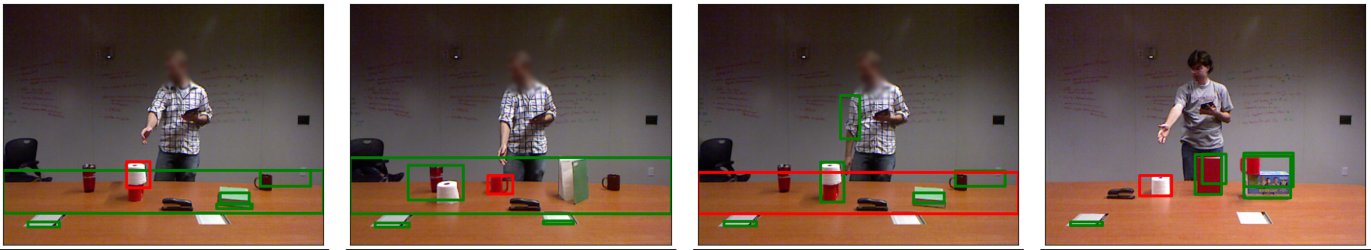
This research was partially sponsored by the National Science Foundation, grant IIS-1218932, and by a gift from Google. The views and conclusions are those only of the authors.

References

[1] A.M. Arsenio. Embodied vision - perceiving objects from actions. In *The 12th IEEE International Workshop on Robot and Human Interactive Communication.*, 2003.

[2] Rehj Cantrell, Matthias Scheutz, Paul Schermerhorn, and Xuan Wu. Robust spoken instruction understanding for HRI. In *Proceedings of HRI*, 2010.

[3] David L. Chen and Raymond J. Mooney. Learning to sportscast: a test of grounded language acquisition. In *Proceedings of ICML*, 2008.



(a)	(b)	(c)	(d)
Orig: toilet papers on the table **LSP+det**: the toilet paper to the right of the green book **LSP**: the toilet paper to the right of the table	**Orig**:the orange mug is on the table **LSP+det**: the orange mug is to the right of the green book **LSP**: the orange mug is to the right of the table	**Orig**: the table beneath the objects **LSP+det**: the table behind the green book **LSP**: the orange 1 to the right of the toilet paper	**Orig**: the toilet paper **LSP+det**: the toilet paper to the right of the red mug **LSP**: the toilet paper to the right of the table

Fig. 4: Example images and corresponding generated natural language descriptions. **Orig** is the original text from the person, **LSP+det** is LSP trained with determiner constraints, and **LSP** is the original model from [14].

[4] David L. Chen and Raymond J. Mooney. Learning to interpret natural language navigation instructions from observations. In *Proceedings of AAAI*, 2011.

[5] James Clarke, Dan Goldwasser, Ming-Wei Chang, and Dan Roth. Driving semantic parsing from the world's response. In *Proceedings of CoNLL*, 2010.

[6] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of CVPR*, 2005.

[7] Juraj Dzifcak, Matthias Scheutz, Chitta Baral, and Paul Schermerhorn. What to do and how to do it: translating natural language directives into temporal and dynamic logic representation for goal management and action execution. In *Proceedings of ICRA*, 2009.

[8] Albert Huang, Edwin Olson, and David Moore. LCM: Lightweight communications and marshalling. In *Proceedings of IROS*, 2010.

[9] Lauri Karttunen. Presupposition and Linguistic Context. *Theoretical Linguistics*, 1:182–194, 1974.

[10] Rohit J. Kate and Raymond J. Mooney. Using string-kernels for learning semantic parsers. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, 2006.

[11] Rohit J. Kate and Raymond J. Mooney. Learning language semantics from ambiguous supervision. In *Proceedings of AAAI*, 2007.

[12] Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas Roy. Toward understanding natural language directions. In *Proceedings of HRI*, 2010.

[13] Hadas Kress-Gazit, Georgios E. Fainekos, and George J. Pappas. Translating structured english to robot controllers. *Advanced Robotics*, 22(12):1343–1359, 2008.

[14] Jayant Krishnamurthy and Thomas Kollar. Jointly learning to parse and perceive: Connecting natural language to the physical world. *Proceedings of the Transactions of the Association for Computational Linguistics*, 2013.

[15] Jayant Krishnamurthy and Tom Mitchell. Weakly supervised training of semantic parsers. In *Proceedings of the Joint Conference on EMNLP and CoNLL*, 2012.

[16] Geert-Jan M. Kruijff, Hendrik Zender, Patric Jensfelt, and Henrik I. Christensen. Situated dialogue and spatial organization: What, where... and why. *International Journal of Advanced Robotic Systems*, 4(1):125–138, 2007.

[17] Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. Inducing probabilistic CCG grammars from logical form with higher-order unification. In *Proceedings of EMNLP*, 2010.

[18] Percy Liang, Michael I. Jordan, and Dan Klein. Learning dependency-based compositional semantics. In *Proceedings of ACL*, 2011.

[19] Matthew MacMahon, Brian Stankiewicz, and Benjamin Kuipers. Walk the talk: connecting language, knowledge, and action in route instructions. In *Proceedings of AAAI*, 2006.

[20] Cynthia Matuszek, Dieter Fox, and Karl Koscher. Following directions using statistical machine translation. In *Proceedings of HRI*, 2010.

[21] Cynthia Matuszek, Nicholas FitzGerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. A joint model of language and perception for grounded attribute learning. In *Proceedings of ICML*, 2012.

[22] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. ROS: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.

[23] Deb Roy, Kai-Yuh Hsiao, and Nikolaos Mavridis. Conversational robots: building blocks for grounding word meaning. In *Proceedings of the HLT-NAACL workshop on learning word meaning from non-linguistic data*, 2003.

[24] Nobuyuki Shimizu and Andrew Haas. Learning to follow navigational route instructions. In *Proceedings of IJCAI*, 2009.

[25] Marjorie Skubic, Dennis Perzanowski, Sam Blisard, Alan Schultz, William Adams, Magda Bugajska, and Derek Brock. Spatial language for human-robot dialogs. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34(2):154–167, 2004.

[26] Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-margin markov networks. In *Proceedings of NIPS*. 2004.

[27] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew Walter, Ashis Banerjee, Seth Teller, and Nicholas Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of AAAI*, 2011.

[28] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of NAACL-HLT*, 2003.

[29] Terry Winograd. *Procedures as a representation for data in a computer program for understanding natural language.* PhD thesis, Massachusetts Institute of Technology, 1970.

[30] John M. Zelle and Raymond J. Mooney. Learning to parse database queries using inductive logic programming. In *Proceedings of AAAI*, 1996.

[31] Luke S. Zettlemoyer and Michael Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of UAI*, 2005.