2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Grounding Spatial Relations for Human-Robot Interaction

Sergio Guadarrama[1], Lorenzo Riano[1], Dave Golland[1], Daniel Göhring[2], Yangqing Jia[1],
Dan Klein[1,2], Pieter Abbeel[1] and Trevor Darrell[1,2]

***Abstract*— We propose a system for human-robot interaction that learns both models for spatial prepositions and for object recognition. Our system grounds the meaning of an input sentence in terms of visual percepts coming from the robot's sensors in order to send an appropriate command to the PR2 or respond to spatial queries. To perform this grounding, the system recognizes the objects in the scene, determines which spatial relations hold between those objects, and semantically parses the input sentence. The proposed system uses the visual and spatial information in conjunction with the semantic parse to interpret statements that refer to objects (nouns), their spatial relationships (prepositions), and to execute commands (actions). The semantic parse is inherently compositional, allowing the robot to understand complex commands that refer to multiple objects and relations such as: "Move the cup close to the robot to the area in front of the plate and behind the tea box". Our system correctly parses 94% of the 210 online test sentences, correctly interprets 91% of the correctly parsed sentences, and correctly executes 89% of the correctly interpreted sentences.**

I. Introduction

In this paper, we present a natural language interface for interacting with a robot that allows users to issue commands and ask queries about the spatial configuration of objects in a shared environment. To accomplish this goal, the robot must interpret the natural language sentence by grounding it in the data streaming from its sensors. Upon understanding the sentence, the robot then must produce an appropriate response via action in the case of a command, or via natural language in the case of a query.

For example, to correctly interpret and execute the command "Pick up the cup that is close to the robot" (see Fig. 1) the system must carry out the following steps: (i) ground the nouns (e.g. "cup") in the sentence to objects in the environment via percepts generated by the robot's sensors; (ii) ground the prepositions (e.g. "close to") in the sentence to relations between objects in the robot's environment; (iii) combine the meanings of the nouns and prepositions to determine the meaning of the command as a whole; and (iv) robustly execute a set of movements (e.g. PICKUP) to accomplish the given task.

In order for a robot to effectively interact with a human in a shared environment, the robot must be able to recognize the objects in the environment as well as be able to understand the spatial relations that hold between these objects. The importance of interpreting spatial relations is evidenced by the long history of research in this area [1], [2], [3], [4], [5]. However, most of the previous work builds models of spatial relations by hand-coding the meanings of the spatial relations rather than learning these meanings from data. One of the conclusions presented in [6] is that a learned model of prepositions can outperform one that is hand-coded. In the present work, we extend the learned spatial relations models presented in [6] to handle a broader range of natural language (see Table I) and to run on a PR2 robot in a real environment such as the one in Fig. 1.[1]

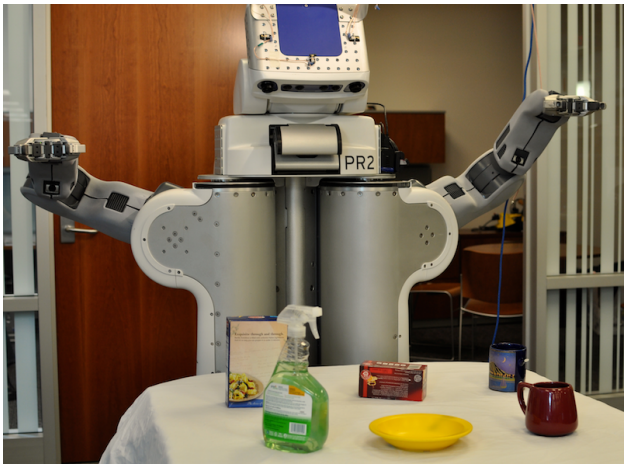

Fig. 1. An example of the visual setting in which the PR2 robot is issued commands and asked queries.

The spatial relations model presented in [6] had several limitations that prevented it from being deployed on an actual robot. First, the model assumed perfect visual information consisting of a virtual $3D$ environment with perfect object segmentation. Second, the model only allowed reference to objects via object ID (e.g. O_3) as opposed to the more natural noun reference ("the cup"). Lastly, the grammar was small and brittle, which caused the system to fail to parse on all but a few carefully constructed expressions. In this work, we extend the model in [6] to address these limitations by building a system that runs on a PR2 robot and interacts with physical objects in the real world.In order to interpret the sentences in Table I, we have built the following modules:

- A vision module that provides grounding between visual percepts and nouns (Section III-B)
- A spatial prepositions module capable of understanding complex $3D$ spatial relationships between objects (Section III-C)

[1] EECS, University of California at Berkeley, Berkeley, CA USA
[2] International Computer Science Institute (ICSI), Berkeley, CA USA
sguada@eecs.berkeley.edu, lorenzo.riano@berkeley.edu, dsg@cs.berkeley.edu, goehring@icsi.berkeley.edu, jiayq@berkeley.edu, klein@cs.berkeley.edu, pabbeel@cs.bekeley.edu, trevor@eecs.berkeley.edu

[1]We will make available demo videos and the supplementary material at `http://rll.berkeley.edu/iros2013grounding`

- A set of actions implemented on a PR2 robot to carry out commands issued in natural language (Section III-D)

We have created an integrated architecture (see Fig. 2), that combines and handles the flow of information of the separate modules. The system is managed by an interface where a user types sentences, and the robot replies either by answering questions or executing commands (see Table I and Figs. 1,3,7). Every sentence is semantically analyzed to determine both the type of query or command as well as the identity of all objects or locations referenced by the sentence. The semantic interpretation depends on the vision module to interpret the nouns and on the prepositions module to interpret the spatial relations present in the sentence. If the sentence issued by the user is interpreted as a command, then the appropriate action and parameters are sent to the robot module. The results from the queries and feedback from the action's execution are finally displayed on the user interface.

Input	Action
"**What is** the object in front of PR2?"	REPLY("A tea box")
"**Which object is** the cup?"	REPLY("It is O_3")
"**Which object is** behind the item that is to the right of the cup?"	REPLY("It is O_7")
"**Which object is** close to the item that is to the left of the green_works?"	REPLY("It is O_6")
"**Point at the area** on the plate."	POINTAT([XYZ])
"**Point to** the object to the left of the tea_box."	POINTTO(O_3)
"**Place** the cup **in the area** behind the plate."	PLACEAT(O_3, [XYZ])
"**Place** the pasta_box **in the area** on the plate."	PLACEAT(O_4, [XYZ])
"**Pick up** the cup that is far from the robot."	PICKUP(O_6)
"**Put down** the cup **in the area** inside the bowl."	PLACEAT(O_6, [XYZ])
"**Pickup** the tea_box in front of the plate."	PICKUP(O_2)
"**Put down** the object **in the area** near to the green_works and far from you."	PLACEAT(O_2, [XYZ])
"**Move** the object that is near to the robot **to the area** far from the robot."	MOVETO(O_2, [XYZ])
"**Move** the cup close to the robot **to the area** in front of the plate and behind the tea_box."	MOVETO(O_3, [XYZ])

TABLE I

EXAMPLES OF SENTENCES HANDLED BY OUR SYSTEM AND THE CORRESPONDING INTERPRETATION.

II. RELATED WORK

Natural language understanding and grounding has been studied since the beginning of artificial intelligence research, and there is a rich literature of related work. Recently, the availability of robotic agents has opened new perspectives in language acquisition and grounding. The seminal work by Steels et. al [7] studied the emergence of language among robots through games. While we retain some of the ideas and concepts, the main difference between our approach and Steels' is that we provide the robot with the vocabulary, whereas in [7] the perceptual categories arise from the agent out of the game strategy. In a similar fashion Roy [8] developed a model that could learn a basic syntax and ground symbols to the sensory data.

Kuipers [9] introduced the idea of Spatial Semantic Hierarchy (SSH), where the environment surrounding the robot is represented at different levels, from geometric to topological. An extension of this work is in [10], where the authors develop a system that follows route instructions. The main contribution is in the automatic synthesis of implicit commands, which significantly improves the robot's performance. However, in contrast with this paper, they use fixed rules rather than learning the spatial relationships from data. In recent work [6], learning these relationships has been shown to be beneficial.

A different approach is to teach language to robots as they perceive their environment. For example, in [11] they present an approach where robots ground lexical knowledge through human-robot dialogues where a robot can ask questions to reduce ambiguity. A more natural approach was presented in [12], where the robot learns words for colors and object instances through physical interaction with its environment. Whereas the language used in [12] only allows direct references, our approach uses complex language that supports spatial reference between objects.

Given the relevance of spatial relations to human-robotic interaction, various models of spatial semantics have been proposed. However, many of these models were either hand-coded [1], [3] or in the case of [2] use a histogram of forces [13] for $2D$ spatial relations. In contrast, we build models of $3D$ spatial relations learned from crowd-sourced data by extending previous work [6].

Some studies consider dynamic spatial relations. In [14], a robot must navigate through an office building, thereby parsing sentences and labeling a map using a probabilistic framework. In [15], a simulated robot must interpret a set of commands to navigate throughout a maze. Our current work focuses mainly on understanding complex spatial relationship between static objects.

Tellex [16] explore language grounding in the context of a robotic forklift that receives commands via natural language. Their system learns parameters for interpreting spatial descriptions, events, and object grounding. In their model, these separate parameters are independent only when conditioned on a semantic parse, and therefore training their model requires annotators to label each sentence with a complex semantic parse. In contrast, we assume a model where the parameters for interpreting spatial descriptions are independent from the object grounding parameters. Hence, instead of requiring structured annotations as in [16], we train on simple categorical annotations, such as the conventional object-label data used in instance recognition settings, which are easier to collect and to generalize.

III. SYSTEM DESCRIPTION

A. Language Module

The language module takes as input a textual, natural language utterance $\mathcal{U}$, which can contain instructions, references to objects either by name or description (e.g., "plate" or "the cup close to the robot"), and descriptions of spatial locations in relation to other objects (e.g., "area behind the plate"). The output of the language module is a command $\mathcal{C}$ to the robot containing the interpretation of the utterance (e.g., PICKUP(O_4)). Interpreting $\mathcal{U}$ into $\mathcal{C}$ happens in three steps: template matching, which decides the coarse

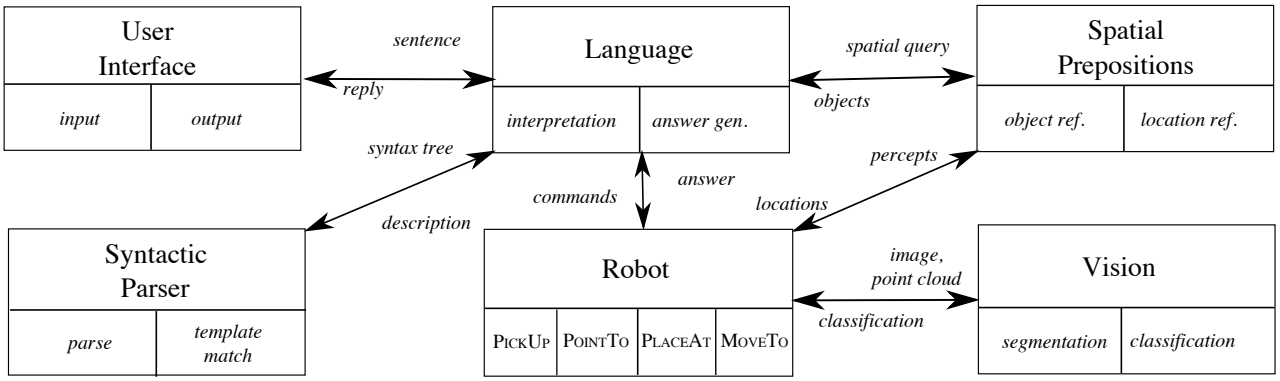

Fig. 2. The architecture of our system showing the interactions between the modules.

form of the sentence; broad syntactic parsing, which analyzes the structure of the sentence; and deep semantic analysis which interprets the linguistic sentence in terms of concepts in the visual setting.

a) Template Matching: First, the utterance $\mathcal{U}$ is matched against a list of manually constructed templates. Each template specifies a set of keywords that must match in $\mathcal{U}$, as well as gaps which capture arbitrary text spans to be analyzed in later steps (a subset are shown in Table I with keywords shown in bold).[2] Each template specifies the query or command as well as which spans of $\mathcal{U}$ correspond to the object descriptions referenced in that command. For example, in the utterance "pick up the cup that is close to the robot", the template would match the keywords "**pick up**" and triggers a PICKUP command to send to the robot. The text spans that must be interpreted as object ids or locations in the environment (such as "the cup that is close to the robot" in our example) are passed to the second step for deeper interpretation.

Although theoretically this template approach structurally limits the supported commands and queries, the approach still covers many of the phenomena present in our data. During evaluation, the templates covered 98% of the tested sentences (see Table III), despite the fact that the humans who generated these sentences were not aware of the exact form of the templates and only knew the general set of actions supported by the robot. We employ the template approach because it closely matches the pattern of language that naturally arises when issuing commands to a robot with a restricted scope of supported actions. Rather than focusing on a broad range of linguistic coverage that extends beyond the capabilities of the robot actions, we focus on deep analysis. In the second and third steps of linguistic interpretation (described below) our system does model recursive descriptions (e.g., "the book on the left of the table on the right of the box"), which are the main linguistic complexity of interest.

b) Broad Syntactic Parsing: In order to robustly support arbitrary references to objects and locations, we parse these descriptions $\mathcal{R}$ with a broad-coverage syntactic parser [17] and then use tree rewrite rules to project the output syntactic parse onto our semantic grammar $\mathcal{G}$:

[noun]	N	$\rightarrow$	`plate` | `cup` | $\cdots$
[preposition]	P	$\rightarrow$	`close_to` | `on` | $\cdots$
[conjunction]	NP	$\rightarrow$	N PP*
[relativization]	PP	$\rightarrow$	P NP

We apply the following tree rewrite rules to normalize the resulting tree into $\mathcal{G}$:[3]

- rename preposition-related POS tags (IN, TO, RB) to P
- crop all subtrees that fall outside $\mathcal{G}$
- merge subtrees from multi-word prepositions into a single node (e.g., "to the left of" into "left")
- to handle typos in the input, we replace unknown prepositions and nouns with those from the lexicons contained in the preposition and vision modules that are closest in edit-distance, provided the distance does not exceed 2

c) Deep Semantic Analysis: The last step of interpretation takes as input a tree $\mathcal{T}$ from our semantic grammar that either refers to a specific object in the robot's environment or a specific $3D$ location. The deep semantic analysis returns the corresponding object id or a list of $3D$ points. For example, in the case of object reference, this step would take the description "the cup that is close to the robot" and return object id $\texttt{O}_4$ (see Fig. 7). We follow the method of probabilistic compositional semantics introduced in [6] to compute a distribution over objects $p(o|\mathcal{R})$ and return the object id that maximizes $\arg\max_o p(o|\mathcal{R})$. Concretely, $\mathcal{T}$ is recursively interpreted to construct a probability distribution over objects. We follow the semantic composition rules presented in [6] at all subtrees except those rooted at N. If the subtree is rooted at N with noun child w, we attain a distribution over objects by leveraging object recognition model (section III-B). We use Bayesian inversion with the uniform prior to transform the object recognition distribution $p(w|o)$ into a distribution over objects given the noun: $p(o|w)$. If the subtree is rooted at PP with children P and NP, the interpretation calls out to the prepositions module (section III-C) to attain the distribution over objects (or $3D$ points, in the case of a location reference) that are in relation P to each of the objects in the recursively computed distribution NP.

[2]These templates were constructed based only on the development data.

[3]These rules were manually generated by analyzing the development data.

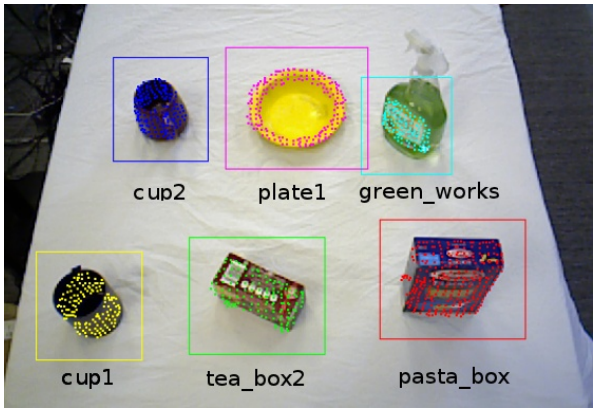

Fig. 3. View of scene in Fig. 1 from the camera perspective. Segmented objects are enframed, corresponding point cloud points are depicted, and object labels are shown.

B. Vision Module

The role of the vision module is twofold: (i) segment the visual input captured by a $3D$ Asus Xtion RGB image and point cloud and (ii) assign a classification score between a noun N and an object id that corresponds to how well the noun describes the object.

1) Training: We trained our object classifier with 50 objects, mainly kitchen and office objects. To obtain training images, we placed the object on a turning table and collected images at a frequency of about 10° per image, collecting around 80 images per object class. Following the idea of [18], we introduced jittering effects to the objects to make the classifier robust against view and perspective changes. Specifically, after we cropped the object inside the bounding box, we randomly transposed, rotated, and scaled the bounding boxes.

2) Segmentation: The $3D$ point cloud captured by the camera is voxelized at a resolution of $1mm$ to reduce the number of points. The points generated from voxelization are transformed from the camera into the robot frame of reference, using the kinematic chain data from the PR2 robot. We fit the plane of the tabletop by applying RANSAC. We constrained the RANSAC by assuming that the table is almost parallel to the ground. All the points that do not belong to the table are clustered to segment out tabletop objects. Noise is reduced by assuming that each object must have a minimum size of $3cm$. The point cloud clusters are subsequently projected into the image to identify image regions to send to the classification module. Fig. 3 shows a segmentation example as described above.

3) Classification: Often, the segmentation component produces well-centered object bounding boxes, allowing us to directly perform object classification on bounding boxes instead of performing object detection, e.g., with a slower sliding window based approach. We apply a state-of-the-art image classification algorithm that use features extracted by a two-level pipeline, (i) the coding level densely extracts local image descriptors, and encodes them into a sparse high-dimensional representation, and (ii) the pooling level aggregates statistics in specific regular grids to provide

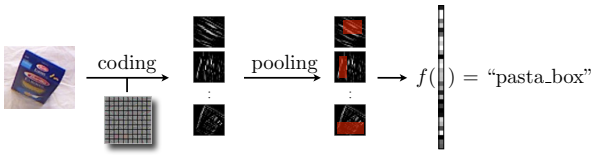

Fig. 4. The classification pipeline adopted to train object classifiers.

invariance to small displacement and distortions. We use a linear SVM to learn the parameters and perform the final classification.

Specifically, we perform feature extraction using the pipeline proposed in [19]. This method has been shown to perform well with small to medium image resolutions,[4] and it is able to use color information (which empirically serves as an important clue in instance recognition). Additionally, the feature extraction pipeline runs at high speed because most of its operations only involve feed-forward, convolution-type operations. To compute features, we resized each bounding box to 32×32 pixels, and densely extracted 6×6 local color patches. We encoded these patches with ZCA whitening followed by a threshold encoding $\alpha = 0.25$ and a codebook of size 200 learned with Orthogonal Matching Pursuit (OMP). The encoded features are max pooled over a 4×4 regular grid, and then fed to a linear SVM to predict the final label of the object. Feature extraction has been carried out in an unsupervised fashion, allowing us to perform easy retraining, should new objects need to be recognized. Fig. 4 illustrates the key components of our pipeline, and we defer to [19] for a detailed description.

C. Spatial Prepositions Module

Given a preposition and landmark object, the prepositions module outputs a distribution over the target objects and $3D$ points that are located in the given preposition in relation to the given landmark object from the robot's point of view.[5]

Following [6], in this work we have focused on the following 11 common spatial prepositions: {`above`, `behind`, `below`, `close_to`, `far_from`, `in_front_of`, `inside_of`, `on`, `to_the_left_of`, `to_the_right_of`, `under`}. We model the meaning of these spatial prepositions using multi-class logistic regression that predicts the identity of a target object (or $3D$ point) g conditioned on a preposition, w, and landmark object, o. The results in [6] suggest that a trained model of spatial prepositions performs better than one that is hard-coded, and so we closely followed the procedure presented therein although we expand the set of features used and propose an hybrid model that choose the appropriate set of features for each spatial preposition.

1) Data Collection: We use the spatial prepositions dataset collected via Amazon's Mechanical Turk (MTK) and introduced in [6]. In this dataset, each annotator is presented with a GoogleSketchup[6] $3D$ model of a room containing a variety of objects arranged in a natural configuration. The annotator is prompted with a preposition w and landmark

[4]Our RGB+depth images have resolution 640×480.
[5]We only consider the robot's point of view for all spatial references.
[6]`http://sketchup.google.com/3dwarehouse/`

object o and must select the target object g that satisfies the relation.

2) Learning the Grounding of Spatial Prepositions: To learn an appropriate model of these prepositions we trained a multi-class logistic regression model with various sets of spatial features computed between the bounding boxes (BBs) of the landmark and target objects. Our model takes the form $p(g|w, o; \theta) \propto \exp \theta^T f(g, w, o)$. We learn the parameters θ by maximizing the log-likelihood of the training data.

Our model includes the following sets of features (inspired by [6], [20], [21], and [22]):

- *Simple Features* are functions only of the center of mass (CM) of the bounding boxes, and are comprised of: the Euclidean distance between the center of mass (CM) and the offsets in X, Y, Z between the CMs.
- *Complex Features* are functions of the relation between the bounding boxes (BBs) and are comprised of: the percentage of overlap of the BBs, the percentage that the target BB is inside the landmark BB, the minimum distance between the BB, and whether or not the target BB is in contact with the landmark BB.
- *Psycholinguistic Features* extend those presented in [21] to $3D$ objects and to all projective prepositions.[7]

Using these sets of features we defined the following models (see Fig. 8 for results):

- *Simple Model* uses simple features.
- *Complex Model* uses simple and complex features.
- *Psycholinguistic Model* uses simple and psycholinguistic features.
- *Combined Model* uses all the features.

To adapt the models learned in the virtual environment to the real-world domain, we used the development set collected with the robot to define a *Hybrid Model* which, for each preposition, selects the best performing model from the four above. Empirically, we found that this method of model selection performed better than others (see section IV-C).

3) Interpreting Object References: The relativization rule in the grammar $\mathcal{G}$ relies on the Hybrid Model of spatial prepositions in order to refer to objects by their physical locations. For example, to interpret the sentence "Pick up the cup that is close to the robot" the language module prompts the prepositions module for a distribution over objects $p(g|\texttt{close\_to}, \texttt{O}_{robot})$, where $\texttt{O}_{robot}$ corresponds to the object id assigned to the robot ($\texttt{O}_1$ in Fig. 7).

4) Interpreting Location References: To interpret references to locations like "the area on plate" or "the area in front of the plate and behind the tea_box" the system returns a distribution over $3D$ points that fall in the described areas.

We simulate placing the target object BB in 1,000 random positions within the boundaries of the table and compute the likelihood of each position given the set of spatial relations expressed in the location reference. We return the best 50 locations, to be filtered by the robot planner to avoid collisions.

[7]above, behind, below, in front of, on, to the left of, to the right of, under

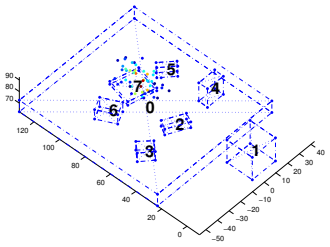

Fig. 5. "on the plate"

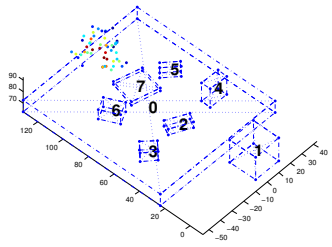

Fig. 6. "behind the plate"

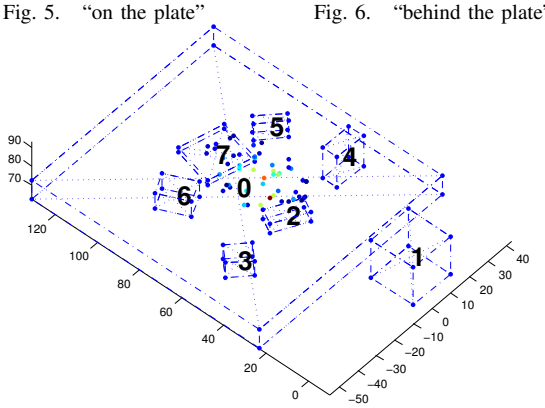

Fig. 7. "in front of the plate and behind the tea_box"

$3D$ points satisfying various spatial relations with regards to the `tea_box` ($\texttt{O}_2$), and `plate` ($\texttt{O}_7$) in the scene depicted in Fig. 1.

For example, "on the plate" refers to the area on $\texttt{O}_7$ (see Fig. 5) and "behind the plate" refers to the area behind $\texttt{O}_7$ (see Fig. 6). While the description "in front of the plate and behind the tea_box" refers to the intersection of the area in front of $\texttt{O}_7$ and behind $\texttt{O}_2$ (see Fig. 7).

D. Robotic Module

Our robotic platform is a mobile manipulator PR2 robot manufactured by Willow Garage.[8] It is two-armed with an omni-directional driving system. Each arm has 7 degrees of freedom. The torso has an additional degree of freedom as it can move vertically. The PR2 has a variety of sensors, among them a tilting laser mounted in the upper body and a $3D$ Asus Xtion camera over a pan-tilt head. During our experiments we used the tilting laser to create a static $3D$ map of the robot's surroundings, and we used the Asus camera to segment and recognize the objects. For planning and executing a collision-free trajectories with the $7DOF$ arms [23], we used a compact $3D$ map [24].

The supported robot actions include:

- PICKUP(O): Pick up object `O` using the algorithm presented in [25].
- POINTTO(O): Point to object `O`.
- POINTAT(`[XYZ]`): Point at location `[XYZ]`.
- PLACEAT(`[XYZ]`): Place down a grasped object in location `[XYZ]`.
- MOVETO(O, `[XYZ]`): Move object `O` into location `[XYZ]`.

PICKUP was implemented by using the object's $3D$ point cloud to compute a good grasping position and by planning a collision-free trajectory to position the gripper for grasping.

[8]`http://www.willowgarage.com`

POINTTO was implemented by moving the robot's gripper so that its tool frame points at the centroid of the object's point cloud. Several candidate gripper positions are uniformly sampled from spheres of various radii around the object. The gripper orientation is chosen to be an orthogonal basis of the pointing vector. The first candidate gripper pose that has a collision-free trajectory is selected for execution.

The PLACEAT action takes as input a candidate list of scored $3D$ points generated by the spatial prepositions module III-C, and places an object held in the gripper at one of these $3D$ points. The robot executes the PLACEAT action using the highest scoring candidate PLACEAT point that yields a collision-free trajectory for the gripper.[9] The exact location for placing takes into account the gripper shape and the object height.

The MOVETO action is implemented by combining PICKUP and PLACEAT.

IV. EXPERIMENTAL RESULTS

A. Experimental Scenario

In our experiments we have collected 290 utterances during development and 210 utterances during the online test. Each utterance is either a command or query issued to the robot by the user in a shared visual setting. As mentioned in section III, we use the development set to design the templates in the language module and perform model selection in the spatial prepositions module. We additionally have collected separate offline datasets in order to train our object classification and preposition models. To train the preposition model, we use the $3D$ virtual environment dataset collected in [6] composed by 43 rooms and consisting of 2,860 tuples of the form (virtual environment, target object, preposition, reference object). However the model presented in this paper uses an expanded set of features and choose the appropriate set of features for each spatial preposition. This new model adapts better to the real environment of the robot.

To train the object classification model, we collect a dataset of 80 pairs of (image, instance label) for each of the 50 possible objects, totalling 4,000 labelled images, that appear in our visual settings. Parameter selection for the remaining components was done using the development set of command/query and visual setting. The result of the robot's execution in response to a command, or linguistic answer in response to a query are evaluated to be either correct or incorrect. We report the performance of each independently trained module (either coverage or accuracy) as well as the accuracy of the overall system on the online test set.

A typical testing scenario is shown in Fig. 1. The dominant feature of the robot's environment is a flat tabletop covered with a set of objects with which the robot will interact. Although we used a white sheet to cover the table, none of our modules depend on a specific background color. Since the objects are segmented using $3D$ point clouds, we assume they are placed at least $2cm$ apart. We further assume all objects are visible by the robot without changing its pan-tilt head configuration, and are reachable by at least one arm without having to move the holonomic base.

[9]To support stacking objects or placing one inside another, we allowed collisions between the gripper-held object and the environment.

B. Vision Results

The object classifier is able to recognize the objects at a high accuracy. In our offline testing, the classifier achieves a 99.8% one-vs-all 10-fold cross validation accuracy over the training set.

In the online testing experiments, the robot was using real data, looking from a 70° angle at the objects on the table. We measured two different accuracies, first the accuracy of the object segmentation, which has to find the patches in the images that contain the objects, and second the classification of the segmented image patch. The object segmentation achieves an accuracy of 94% when we evaluate the segmentation over the objects contained in the online testing. Wrong object segmentations were usually caused by reflective object surfaces, e.g., reflective pans. Here, the Asus camera did not perceive $3D$-points in larger areas of the objects. Typically, in these cases the object segmentation identified more than one object.

The object classification achieved an accuracy of 91.7% during online testing. Only correctly segmented object areas were considered for this classification experiment. However, in this work we are more interested in a selection task than in a classification one. In the selection task, the goal is to select the target object o_w described by some words w: $o_w = \text{argmax}_o(p(o|w))$, while in the classification task the goal is to label w_o a given object o: $w_o = \text{argmax}_w(p(w|o))$. When the task is to select one object among the ones on the table, the selection accuracy is 97%. The selection accuracy is higher that the classification accuracy mainly because the model has to choose between 5-6 objects, while for the classification the model has to choose between 50 labels.

Segmentation	Classification	Selection
94%	91.7%	97%

TABLE II
VISION RESULTS IN THE ONLINE TEST

C. Spatial Prepositions Results

In this section we present the results of testing the spatial preposition module independently of the rest of the architecture.

Given a landmark and a spatial preposition, the module predicts a target object. We reported how often the predicted object matches human judgment for the same task. We extracted 300 (landmark, spatial prepositions)-pairs from our online test set, and asked 3 different people to select the set of valid targets and pick a "best answer" from among that set. The ground truth is defined by majority vote.

As can be seen in Fig. 8, the random baseline for selecting the best answer is 14%, while the inter-annotator accuracy ("humans") is 85%. Human agreement is below 100% due to

the inherent difficulty in selecting the best answer for some ambiguous prepositions (e.g. close to, far from).

During our experiments we have found that the intrinsic ambiguity of some of the queries is a significant source of errors. For example, reaching a consensus on the answer to "which object is far from the robot" in Fig. 3 is challenging. However, if we allow more than one answer to be correct (e.g. all the objects in the second row are considered far from the robot) then the spatial prepositions module's error rate reduces greatly (as can be seen in Fig. 8).

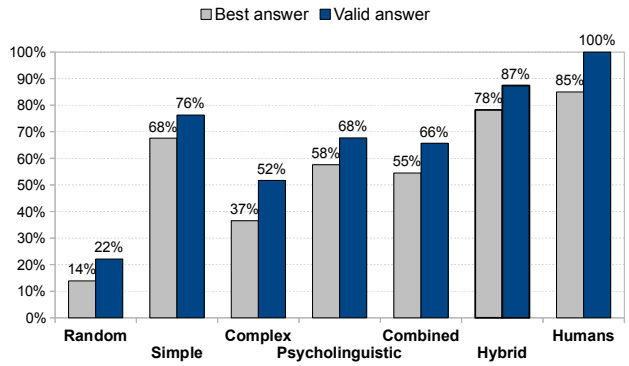

Fig. 8. Spatial Prepositions results

The best results for the spatial preposition module were obtained with the Hybrid Model, which independently chooses the best model (among Simple, Complex, Psycholinguistic, and Combined) for each preposition based on a validation set of 100 triples (landmark, spatial preposition, target) extracted from the development set.

D. Overall Results and Error Analysis

To evaluate the performance of the whole system we measured the accuracy of each of the needed steps to correctly interpret and answer/execute a question/command given in natural language by the user. The results for each of these steps on the online test set (210 sentences) are presented in Table III:[10]

- Template Matching: Percentage of sentences that match one of the predefined templates. In this case the main source of errors is the user misspelling of words (a).
- Grammar Coverage: Percentage of sentences that the Language Module can parse after the template matching succeed. In this case the main source of errors are failures in the tree normalization process (b,c).
- Noun Interpretation: Percentage of nouns that the language module can generate a valid answer using the classification results from the vision module. In this case the main source of errors is the wrong segmentation (d,e).
- Preposition Interpretation: Percentage of spatial prepositions that the language module can generate a valid answer using the predictions results from the spatial preposition module. In this case the main source of error is the ambiguity of the target referred by the spatial prepositions model (f,g).
- Sentence Interpretation: Percentage of parsed sentences that the system can correctly interpret by combining the results from the noun and preposition interpretations according to the syntactically normalized tree. In this case the main source of error is the inability of the system to choose the best answer within the valid set of answers (e,g,h).
- Execution: Percentage of interpreted sentences that the robot can execute correctly (when the input is a command). The main sources of errors are non-reachable poses in the robot's configuration space and collisions during placing (i,j).

[10]In parentheses we have included references to examples of errors from Table IV.

Template Matching	98%
Grammar Coverage	94%
Noun Interpretation	97%
Preposition Interpretation	95%
Sentence Interpretation	**91%**
Valid Execution	**89%**

TABLE III
OVERALL RESULTS IN THE ONLINE TEST

(a) *Poit* the left of the bowl (Template)
(b) **Which object is** behind the item which is to the left of the cup? (Grammar)
(c) **Pickup** the cup near to PR2 (Grammar)
(d) **What is** to the left of the pan (Nouns)
(e) **Place** the tea_box **in the area** near to the coffe_mate (Nouns)
(f) **Point at** the object on the left of the green_works (Prepositions)
(g) **Point to** the object to the left of the tea_box (Prepositions)
(h) **Which object is** to the left of the mug and to the right of the cup? (Sentence)
(i) **Pick up** the pan (Execution)
(j) **Move** the cup in front of the pan **into the area** on the clock (Execution)

TABLE IV
EXAMPLES OF FAILED SENTENCES

V. DISCUSSION AND CONCLUSIONS

The contribution of this paper is an extension of [6] to a real robotic system with sensor-driven perception for grounding nouns and spatial relations. It is noteworthy that, while the data used for training the spatial prepositions module has been acquired via a virtual world, the model has proven general enough to yield acceptable performance in a real robotics scenario.

Our results in Table III show that the overall system is capable of executing complex commands issued in natural language that grounds into robotic percepts. Although the modules in our system are trained in isolation, a correct interpretation requires that they all work together.

Our results suggest that stronger integration between modules is a fruitful avenue for reducing interpretation errors. For instance, the combination of the noun interpretation with preposition interpretation helps to reduce ambiguity in the descriptions. For example, in Fig. 3 there are two cups and three objects close to the robot, and therefore the commands "pick up the cup" and "pick up the object close to the robot" are ambiguous. However, the command "pick up the cup close to the robot" helps determine the

relevant cup. This capability could be used to enable multiple grounding sources for the objects. For example, assertions like "the object in front of the plate is a tea_box" or "the green_works is the object behind the pasta_box" can be used to teach the system new labels and spatial relations via linguistic input.

Stronger integration between components within a single module can also help reduce errors. Currently, the language interpretation module works in a feed-forward, pipelined approach: first templates are used for coarse language matching, next text spans are parsed and projected into a small semantic grammar, and then the semantic trees are interpreted. A failure in one layer will propagate to subsequent layers. In future work, we plan to refactor our model to remove this limitation by performing joint inference so that decisions are made using information from all steps of the process. We eventually plan to extend the joint model to incorporate the computer vision and spatial prepositions module, so that all components share information more directly in order to help each other make decisions.

Our system correctly interprets many of the input sentences; therefore, in addition to reducing the errors, we are interested in extending the system to handle increased complexity. We are working towards enabling the robot to understand and execute more complex sentences, including actions that will require a degree of planning or actions that unfold over long periods of time. Moreover, we are working to grow the lexicon beyond our initial set of nouns and prepositions. We are additionally working on enabling the robot to operate in more generic, non-tabletop scenarios.

VI. Acknowledgments

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) in the United States as part of Activity E within the Broad Operational Language Translation (BOLT) program.

The work of S. Guadarrama is supported by the Spanish Deparment of Science and Education (ref. EX2009-1073), including EDRF/FEDER funding.

References

[1] W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "Experiences with an interactive museum tour-guide robot," Artificial Intelligence, vol. 114, no. 1, pp. 3–55, 1999.

[2] M. Skubic, D. Perzanowski, S. Blisard, A. Schultz, W. Adams, M. Bugajska, and D. Brock, "Spatial language for human-robot dialogs," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 34, no. 2, pp. 154–167, 2004.

[3] R. Moratz and T. Tenbrink, "Spatial reference in linguistic human-robot interaction: Iterative, empirically supported development of a model of projective relations," Spatial Cognition and Computation, vol. 6, no. 1, pp. 63–107, 2006.

[4] J. D. Kelleher, G.-J. M. Kruijff, and F. J. Costello, "Proximity in context: An empirically grounded computational model of proximity for processing topological spatial expressions," in Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics. Sydney, Australia: Association for Computational Linguistics, July 2006, pp. 745–752. [Online]. Available: http://www.aclweb.org/anthology/P06-1094

[5] H. Zender, G.-J. M. Kruijff, and I. Kruijff-Korbayová, "Situated resolution and generation of spatial referring expressions for robotic assistants," in IJCAI, 2009, pp. 1604–1609.

[6] D. Golland, P. Liang, and D. Klein, "A Game-Theoretic Approach to Generating Spatial Descriptions," in Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. Cambridge, MA: Association for Computational Linguistics, October 2010, pp. 410–419.

[7] L. Steels and P. Vogt, "Grounding adaptive language games in robotic agents," Proceedings of the fourth european conference on artificial life, vol. 97, 1997.

[8] D. Roy, "Learning visually grounded words and syntax for a scene description task," Computer Speech & Language, vol. 16, no. 3, pp. 353–385, 2002.

[9] B. Kuipers, "The spatial semantic hierarchy," Artificial Intelligence, vol. 119, no. 1, pp. 191–233, 2000.

[10] M. MacMahon, B. Stankiewicz, and B. Kuipers, "Walk the talk: Connecting language, knowledge, and action in route instructions," Def, vol. 2, no. 6, p. 4, 2006.

[11] M. Nakano, N. Iwahashi, T. Nagai, T. Sumii, X. Zuo, R. Taguchi, T. Nose, A. Mizutani, T. Nakamura, M. Attamim, H. Narimatsu, K. Funakoshi, and Y. Hasegawa, "Grounding new words on the physical world in multi-domain human-robot dialogues," in AAAI Fall Symposia on Dialog with Robots, 2010.

[12] N. Iwahashi, K. Sugiura, R. Taguchi, T. Nagai, and T. Taniguchi, "Robots that learn to communicate: A developmental approach to personally and physically situated human-robot conversations," in AAAI Fall Symposia on Dialog with Robots, 2010.

[13] P. Matsakis and L. Wendling, "A new way to represent the relative position between areal objects," in IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 21, no. 7, 1999, pp. 634–643.

[14] C. Matuszek, D. Fox, and K. Koscher, "Following directions using statistical machine translation," in Proceedings ACM/IEEE Int'l Conf. on Human-Robot Interaction (HRI), 2010.

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

[16] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. Teller, and N. Roy, "Understanding Natural Language Commands for Robotic Navigation and Mobile Manipulation," in Proceedings Of The National Conference On Artificial Intelligence, no. Aaai, 2011.

[17] S. Petrov and D. Klein, "Improved Inference for Unlexicalized Parsing," in Proceedings of HLT-NAACL. Rochester, New York: Association for Computational Linguistics, April 2007, pp. 404–411.

[18] S. Winder and M. Brown, "Learning local image descriptors," in CVPR, IEEE Conference on, 2007.

[19] A. Coates, H. Lee, and A. Ng, "An analysis of single-layer networks in unsupervised feature learning," in AISTATS, 2010.

[20] S. Guadarrama and D. Pancho, "Using soft constraints to interpret descriptions of shapes," in Semantic Computing (ICSC), 2010 IEEE Fourth International Conference on. IEEE, 2010, pp. 341–348.

[21] T. Regier and L. Carlson, "Grounding spatial language in perception: an empirical and computational investigation." Journal of Experimental Psychology: General, vol. 130, no. 2, p. 273, 2001.

[22] P. Gorniak and D. Roy, "Grounded semantic composition for visual scenes," J. Artif. Intell. Res. (JAIR), vol. 21, pp. 429–470, 2004.

[23] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," IEEE Robotics & Automation Magazine, 2012, to appear. [Online]. Available: http://ompl.kavrakilab.org

[24] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems," in Proc. of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation, Anchorage, AK, USA, May 2010, software available at http://octomap.sf.net/. [Online]. Available: http://octomap.sf.net/

[25] K. Hsiao, S. Chitta, M. Ciocarlie, and E. Jones, "Contact-reactive grasping of objects with partial shape information," in Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on. IEEE, 2010, pp. 1228–1235.