Driving Under the Influence (of Language)

Daniel Paul Barrett, Student Member, IEEE, Scott Alan Bronikowski, Student Member, IEEE, Haonan Yu, Student Member, IEEE, and Jeffrey Mark Siskind, Senior Member, IEEE

Abstract—We present a unified framework which supports grounding natural-language semantics in robotic driving. This framework supports acquisition (learning grounded meanings of nouns and prepositions from human sentential annotation of robotic driving paths), generation (using such acquired meanings to generate sentential description of new robotic driving paths), and comprehension (using such acquired meanings to support automated driving to accomplish navigational goals specified in natural language). We evaluate the performance of these three tasks by having independent human judges rate the semantic fidelity of the sentences associated with paths. Overall, machine performance is 74.9%, while the performance of human annotators is 83.8%.

Index Terms—Natural Language Robot Interaction and Control, Cognitive Human-Robot Interaction, Wheeled Robots, Learning and Adaptive Systems, Nonholonomic Motion Planning

I. INTRODUCTION

With recent advances in machine perception and robotic automation, it becomes increasingly relevant and important to allow machines to interact with humans in natural language in a grounded fashion, where the language refers to actual things and activities in the world. Here, we present our efforts to automatically drive—and learn to drive—a mobile robot under natural-language command. Our contribution is summarized in Fig. 1. A human teleoperator is given a set of sentential instructions designating robot paths. The operator then drives a mobile robot under radio control according to these instructions through a variety of floorplans. The robot uses onboard odometry and inertial guidance sensors to determine its location in real time and saves traces of the driving paths to log files. Visualizations of these paths are then placed on Amazon Mechanical Turk (AMT), where anonymous workers are asked to provide sentential descriptions. From a training corpus of these paths paired with their corresponding sentential descriptions and floorplan specifications, our system automatically learns the meanings of nouns that refer to objects in the floorplan and prepositions that describe both the spatial relations between floorplan objects and between such objects and the robot path. With such learned meanings, the robot can then generate sentential descriptions of new driving activity undertaken by a teleoperator. Moreover, instead of manually controlling the robot through teleoperation, one can issue the robot natural-language commands which induce fully automatic driving to satisfy the path specified in the natural-language command.

We have conducted experiments with an actual radio-controlled robot that demonstrate all three of these modes of operation: acquisition, generation, and comprehension. We demonstrate successful completion of all three of these tasks on hundreds of driving examples. We evaluate the fidelity of the sentential descriptions produced automatically in response to manual driving and the fidelity of the driving paths induced automatically to fulfill natural-language commands, by presenting the pairs of sentences together with the associated paths to anonymous human judges on AMT. For machine generated results overall, the average “sentence correctness” (the degree to which the sentence is true of the path) reported is 74.2%, the average “path completeness” (the degree to which the path fully covers the sentence) reported is 76.0%, and the average “sentence completeness” (the degree to which the sentence fully covers the path) reported is 74.5%, for an average of 74.9%.

II. RELATED WORK

We know of no other work which presents a physical robot which learns word meanings from driven paths paired with sentences, uses these learned meanings to generate sentential descriptions of driven paths, and automatically plans and physically drives paths satisfying sentential descriptions.

Fig. 2 compares the properties of the work reported in this manuscript with the work reported in twenty recent related papers which are further discussed below.

While there is other work which learns the meanings of words in the context of description of navigation paths, these systems operate only within discrete simulation, as they utilize the internal representation of the simulation to obtain discrete symbolic primitives [1]–[8]. They have a small space of possible robot actions, positions, and states which are represented in terms of symbolic primitives like TURN LEFT, TURN RIGHT, and MOVE FORWARD $N$ STEPS (e.g., [5], [6]), or DRIVE TO LOCATION 1 and PICK UP PALLET 1 (e.g., [8]). Thus, they take a sequence of primitives like \{DRIVE TO LOCATION 1; PICK UP PALLET 1\} and a sentence like go to the pallet and pick it up and learn that the word pallet maps to the primitive PALLET, that the phrase pick up maps to the primitive PICK UP, and that the phrase go to X means DRIVE TO LOCATION X.

We solve a more difficult problem. Our robot and environment are in the continuous physical world and can take an uncountably infinite number of configurations. Our input is a set of sentences matched with robot paths, which are sequences of points in the real 2D Cartesian plane, densely sampled in time. Not all points in the path correspond to words in the sentences; multiple (often undescribed) relationships can be true of any point, and the correspondence between described relationships and path points is unknown. Furthermore, our
The robot went behind the cone and then turned around and went further behind the cone to the right of the chair.

Fig. 1. (left) A human drives the mobile robot through paths according to sentential instruction while odometry reconstructs the robot’s paths. Natural-language descriptions of these paths are obtained from AMT. This allows the robot to learn the meanings of the nouns and prepositions. Hand-designed word models are shown here for illustration; actual learned word models are shown in Fig. 14. Note that the distributions are uniform in velocity angle (bottom row) for left of, right of, in front of, and behind and in position angle (top row) for towards and away from. These learned meanings support generation of English descriptions of new paths driven by teleoperation (top right) and autonomous driving of paths that meet navigational goals specified in English descriptions (bottom right).

<table>
<thead>
<tr>
<th>Physical Robot?</th>
<th>N/A (does not learn)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learn word meanings?</td>
<td>N/A (does not learn)</td>
</tr>
<tr>
<td>Generate descriptions of paths?</td>
<td>N/A (does not learn)</td>
</tr>
<tr>
<td>Generate then drive or follow paths from descriptions?</td>
<td>N/A (does not learn)</td>
</tr>
<tr>
<td>Noisy Data?</td>
<td>N/A (does not learn)</td>
</tr>
<tr>
<td>Domain</td>
<td>Continuous or Discrete</td>
</tr>
<tr>
<td>Environment?</td>
<td>Real or Virtual</td>
</tr>
<tr>
<td>Language Corpus Size (# of words)</td>
<td>12,616 total</td>
</tr>
<tr>
<td>Unspecified in N/A</td>
<td>247 unique</td>
</tr>
<tr>
<td>Learned Lexicon Size (# of words)</td>
<td>12,616 total</td>
</tr>
<tr>
<td>Nouns, Prepositions, or Adjectives</td>
<td>247 unique</td>
</tr>
<tr>
<td>Unspecified in N/A</td>
<td>12</td>
</tr>
<tr>
<td>Learns without mutual annotation?</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Notes:
1. Does not learn individual word meanings, instead maps from NL phrases to compound action specifications which apply contextual cues of the phrase to the parameters of one of a small number of predetermined actions.
2. Uniqueness of such is not stated.
3. Requires hand-drawn examples of paths that depict spatial relations as well as hand-grounded locations of objects.
4. Requires the manual grounding of all words to specific objects or relationships (prepositions) in the training corpus.
5. Uses hand-coded semantic fields and nouns which are manually grounded to map data.
6. All experiments are done in simulation, but some environment maps are created from SLAM data from a physical robot.
7. Words in prepositions and noun phrases to manually-annotated object groundings.
8. Learning system uses a Robot Manager module that abstracts the noisy, continuous data into a discrete, symbolic form.
9. Language corpus consists of 189 unique sentences.
10. Requires natural-language commands to be hand-segmented into individual movement phrases and then manually annotated in RCL for training.
11. Uses same dataset as [1] and [5], and thus needs the same manual parsing of NL phrases to one of a small number of movement, location, or logic terms in their Robot Control Language.

Authors collect dataset from a SLAM-equipped mobile robot, then create from this a discrete grid-map (a graph with vertices and edges) which is used to conduct experiments in simulation.

Uses hand-drawn examples of paths that depict spatial relations as well as hand-grounded locations of objects.

Requires the manual grounding of all words to specific objects (nouns) or relationships (prepositions) in the training corpus.

Uses hand-coded semantic fields and nouns which are manually grounded to map data.

Learns mappings from natural-language noun phrases to manually-annotated object groundings.

Utilizes training data from [1] that has been manually parsed and aligned so that each sentence corresponds to a single predetermined action.

Uses natural-language commands to be hand-segmented into individual movement phrases and then manually annotated in RCL for training.

Language corpus consists of 189 unique sentences.

Requires the manual grounding of all words to specific objects (nouns) or relationships (prepositions) in the training corpus.

Uses hand-coded semantic fields and nouns which are manually grounded to map data.

Fig. 2. A comparison of the properties of the work reported in this manuscript with that reported in twenty recent related papers. Unless otherwise noted, the green and red boxes mean yes and no respectively.
system does not require the additional manually annotated data upon which much of the previous work depends.

MacMahon et al. [1] introduce a software agent that follows natural-language route instructions in a discrete grid-based virtual indoor environment. This system does not learn individual word meanings. Instead, it maps from natural-language phrases to what they call a compound action specification, which applies the linguistic contextual cues of the instruction to the parameters of one of the four predetermined actions that their system can take (TURN, TRAVEL, VERIFY, and DECLARE-GOAL). This system also requires manually annotated data, in the form of hand-verified parse trees of the natural-language route instructions, as input.

Kollar et al. [2] present a system for natural-language direction following that operates in simulation on a discrete virtual environment derived from SLAM data collected by a physical robot. This system requires two different types of manual annotation. To locate objects (nouns), it takes a manually seeded map of the names and locations of such objects. To learn the meanings of prepositions, this system requires a corpus of hand-drawn examples of paths, both positive and negative, that depict such prepositions.

Matuszek et al. [3] also do experiments in simulation on a discrete topological grid map derived from SLAM data from a physical robot, similar to Kollar et al. [2], although their maps are segmented using the approach in Friedman et al. [21]. They focus on using statistical machine translation to map from imperative natural-language directions (e.g., take the second left) to a path description language which directs the motion of their simulated robot, taking advantage of the geometry of the map to constrain the combinatorics of possible actions at a given location. They learn no individual word meanings, as we do; instead, they learn the mappings between phrases, such as the previous example, and the sequence of simulated robot commands that follow such an instruction. Their system is not capable of handling natural-language descriptions of objects or other landmarks that appear in the environment. Finally, their evaluation is on a small corpus of only fourteen sets of route instructions and uses an oracle to evaluate whether the simulated robot reaches the correct destination by the intended route.

Tellex et al. [4] describe a system for understanding natural-language commands prescribing navigation and limited manipulation in discrete simulated environments. Like our system, this system utilizes natural-language commands collected from AMT. However, this system is only capable of learning word meanings, represented as the words’ groundings within the environment, not using the learned word meanings for subsequent tasks. To do so, it requires manual annotation of the groundings of all words in the training sentences to specific objects (nouns) and relationships (prepositions) in the training data. It is incapable of generating either sentential descriptions from given paths, or paths from given descriptions.

Chen & Mooney [5] present a system that learns to interpret natural-language navigation instructions by mapping them to executable plans. This system uses the discrete grid-based virtual environments, data, and simulation execution module developed and used by MacMahon et al. [1]. Like that system [1], this system does not directly learn word meanings, but rather it maps phrases to compound action specifications which in turn map to the parameters of one of three predetermined actions (TURN, TRAVEL, and VERIFY). Additionally, Chen & Mooney manually parse and align the data from MacMahon et al. [1] to form training data in which each sentence in a human-generated instruction is paired with a corresponding predetermined action.

Matuszek et al. [6] seek to learn a semantic parsing model which takes natural-language instructions and produces robot-executable commands in what they call Robot Command Language (RCL). They learn a distribution over possible RCL sequences through the use of natural-language commands, which are first hand-segmented into discrete movement phrases and then paired with expert-created annotations of such in RCL. This is similar to both MacMahon et al. [1] and Chen & Mooney [5]; all learn the mappings from phrases to commands rather than representations of the meanings of individual words. Finally, the authors test their system by simulating the movement of a robot within a discrete grid-based virtual indoor environment and making a binary judgment on whether or not the simulated robot reaches the intended destination by the desired route.

Artzi & Zettlemoyer [7] learn a semantic parser as well as individual word meanings, represented as the grounding of objects or relationships within their test environment. They utilize the same dataset as MacMahon et al. [1] and Chen & Mooney [5] and thus operate in simulation within the same discrete grid-based virtual environment used in the work described in those papers; however, they hand-filter the dataset to include only correct sentence-trace pairs. Similarly, this work also requires a training set in which natural-language instructions are manually paired with executable action sequences.

Tellex et al. [8] present a system that learns grounded word meanings from a training corpus consisting of natural-language commands paired with video or log data depicting such actions, similar to our acquisition task (Section V-A). This system also follows paths derived from sentential descriptions, much like our comprehension task (Section V-C). However, this system operates only in simulation in a grid-based virtual environment with a discrete space of possible robot actions. Furthermore, this system also requires manual annotation of sentence parses and object groundings for each natural-language command in the training corpus. The system then learns the mappings from natural-language noun phrases to these object groundings.

There has been work on learning in the context of language and mobile robot navigation using a physical robot (e.g., [9], [10]), but none of these do all three of the tasks (acquisition, generation, and comprehension) which we do.

Dobnik et al. [9] have an actual robot but only learn to classify prepositional phrases like A is near B from robot paths paired with such phrases that have hand-grounded nouns. They neither generate sentences, nor automatically drive paths. Our system can do both of these, as well as learn meanings for both nouns and prepositions.

Lauria et al. [10] use a collected corpus of high-level route instructions given by naïve users to a small robot operating
in a miniaturized outdoor setting in order to learn symbolic representations of motion tasks that can generalize such instructions. The system described therein assumes the use of pre-programmed primitives which encode the basic sensory-motor actions of the robot. The authors manually analyze their collected corpus and extract a set of 14 primitives, from the simple (TURN RIGHT) to the complex (TAKE THE NTH TURN AFTER X), which they ground to robot sensory-motor actions by hand. Their learning occurs at a higher level of abstraction than ours, focusing on finding the mappings from natural-language route instructions to sequences of these hand-crafted action primitives.

There is also recent work on the topic of natural-language interaction with robots (e.g., [11]–[16]), both within and outside the realm of robotic navigation. However, such work does not involve any learning.

Teller et al. [11] present a robotic forklift which accepts speech and pen-based input, operates in semi-structured outdoor environments, and has a robust sensing architecture which allows it to operate in close proximity to humans. This system uses an off-the-shelf speech recognizer to turn spoken utterances into text. Such utterances are limited to a small set of phrases that direct movement. The system’s responses to the text derived from these utterances is based on pre-defined knowledge of locations in the environment and hand-programmed mappings between the words of an utterance and robotic actions required to fulfill such. This system does no learning and cannot respond to commands for which it does not possess manually-defined primitives.

Koller et al. [12] shows similarity to our generation task (Section V-B) in that while we generate sentences that describe a robot path, their systems provide instructions to guide human users through virtual game worlds. Unlike our system, however, the systems described therein operate in a virtual environment, which gives them a complete and unambiguous discrete symbolic representation of the world. Additionally, their systems do no learning and instead rely on hand-coded mappings between words and actions in order to generate instructions.

Harris et al. [13] present a dialog system in which a single human interacts verbally with multiple robots in a simulation of a treasure-hunt scenario. Human participants are given a map of a maze and must direct robots from unknown starting locations to a destination using only imperative commands like MOVE NORTH 5 METERS and responses to queries like REPORT. Like many other simulation systems, this system uses a discrete grid-based virtual environment with a complete and unambiguous internal symbolic representation. This system does no learning and is instead focused on disambiguation of instructions from a single human sender to multiple robot receivers.

Marge et al. [14] describe TEAMTALK, a human-robot interface which allows verbal interaction between multiple people and multiple robots in a virtual treasure-hunt scenario similar to Harris et al. [13]. Its primary goal is to construct a policy which is able to handle dynamic and asynchronous conversation between a group of four or more humans and robots. Like Harris et al. [13], TEAMTALK uses a discrete grid-based simulation environment with a symbolic representation. TEAMTALK also does no learning.

Pappu & Rudnicky [15] collect and present the NAVIGATI corpus, a dataset of human-generated route instructions in an indoor environment. Their goal is to analyze the corpus and construct a grammar that would generalize to other navigational corpora. The authors focus solely on the linguistic content of the instructions and neither learn word meanings nor control robots, simulated or real, via language.

Fasola & Mataric [16] investigate methods for enabling service robots to better understand spatial language from non-expert users. To do this, they use semantic fields to represent prepositions. Their system requires prepositions to be manually encoded as such. It also requires the grounding of nouns to map data. With such human-generated knowledge, the system is able to perform path planning in a variety of discrete simulation environments. This is similar to our comprehension task (Section V-C), although our system operates in the continuous physical world using knowledge learned in the acquisition task (Section V-A). Their system does no learning.

There is also work which performs learning in the context of language and robotics, but not navigation (e.g., [17]–[20]).

McGuire et al. [17] use gesture recognition and speech commands to interact with a robot arm by human demonstration. This allows a user to train the robot arm to associate a natural-language command with a specific manipulation task. However, this system is unable to respond to a natural-language command for which it is not specifically programmed.

Doshi & Roy [18] present a dialogue management system designed to learn to overcome the limitations of noisy speech recognition within the paradigm of a robotic wheelchair. All testing of this learning approach is done in simulation using a small number of discrete states. As their focus is on dialogue management, they do no navigation; they assume that if their system is able to determine the correct goal state, it is also able to navigate to such via a manually-programmed route.

Matuszek et al. [19] propose a joint framework to learn linguistic and perceptual, primarily visual, models for grounding language to physical objects in a static scene. Their system trains color and shape classifiers on objects segmented from images captured with an RGB-D camera and then learns the mapping between such classifiers and linguistic descriptors in the form of adjectives (i.e., green) and nouns (i.e., triangle). While both this work and ours ground language in robotic perception, the modalities of such perception are so different as to preclude direct comparison. Our system perceives its environment via mechanical sensors such as wheel encoders and an IMU (Section III), while their system employs computer-vision techniques on camera data. Our current work does not utilize any vision data, although we intend to do so in the future. Furthermore, while their system learns nouns and static adjectives (colors) that describe such, our system learns nouns as well as prepositions that describe noun properties which are both static (spatial relations between stationary objects) and dynamic (spatial relations between a stationary object and our mobile robot).

She et al. [20] teach a physical manipulator arm to interact
with objects through student-teacher dialogue. Their work connects high-level symbolic representations of language with low-level sensorimotor representations internal to the robot. Like Matuszek et al. [19], their system perceives the world primarily through visual means, which is starkly different from the mechanical perception that our system uses. Additionally, their system learns manipulation tasks through interactive dialogue with a human, while our system learns individual word meanings from a fixed training corpus.

III. OUR MOBILE ROBOT

All experiments were performed on a custom mobile robot (Fig. 3). This robot can be driven by a human teleoperator or drive itself automatically to accomplish specified navigational goals. During all operation, robot localization is performed onboard the robot in real time via an Extended Kalman Filter [23] with odometry from shaft encoders on the wheels and inertial guidance from an IMU.

Due to sensor noise and mechanical factors such as wheel slippage, this localization is noisy, but generally within 20cm of the actual location. The video feed, localization, and all sensor and actuator data are logged in a time-stamped format. When conducting experiments on acquisition and generation, a human teleoperator drives the robot along a variety of paths in a variety of floorplans. The paths recovered from localization support acquisition and generation. When conducting experiments on comprehension, a path is first planned automatically, then the robot follows this planned path by comparing the new odometry gathered in real time with the planned path and controlling the wheels accordingly.

The use of an actual robot with noisy real-world sensor data increases the difficulty of these tasks when compared to work which occurs in simulation. The noisy robot position is densely sampled in the continuous domain. For acquisition and generation, this adds an additional layer of uncertainty, as the correspondence between individual points in the robot path and the phrases of a sentence is unknown. For comprehension, the robot must find a path by choosing waypoints in the continuous domain that both maximally satisfy the meaning of the instructions and avoid collision with objects. It does not rely on a small set of discrete locations with associated symbolic commands like goto(table), as many (simulated and robotic) systems do.

IV. EXTRACTING MEANING FROM A SENTENCE

This paper concerns itself with semantics, not syntax, and only addresses issues relating to the grounding of word meanings. We represent the meaning of a sentence describing a robot path as a sequence of graphical models composed of probability distributions corresponding to word meanings. Each such graphical model represents the meaning of a clause describing the path at a particular point in time. Each graphical model is a factorized joint distribution over a set of variables: a path variable, which is a pair of 2D vectors representing the position and velocity of the robot at that particular time, and a set of floorplan variables, which are labeled 2D Cartesian coordinates representing the class and position of each object in a floorplan.

A. Constructing graphical models from a sentence

We automatically generate such a sequence of graphical models directly from a sentence. Fig. 4 illustrates this process. Each noun induces a floorplan variable with a univariate distribution over possible object labels that it may take. Each preposition induces a joint distribution between its target and referent objects. Prepositions may be used adverbially to describe the motion of the robot in relation to objects in the floorplan, or adjectively to describe the static position of an object relative to other objects.

The task of constructing a sequence of graphical models from a sentence has three parts: breaking the sentence into its temporally sequential parts, identifying nouns and prepositions, and determining the arguments to each preposition. Off-the-shelf semantic parsers such as the Stanford parser [24] produce largely erroneous parse trees on our corpus of sentences, and thus cannot be used directly for this purpose. We do, however, use the Stanford parser to perform part-of-speech tagging. Fasola & Mataric [16] similarly use the Stanford parser solely for part-of-speech tagging.

To break a sentence into its temporal segments, we identify all verbs that do not immediately follow a WH-determiner (i.e., which, that, etc.), as well as adverbial transition words (i.e., then) and use them as the segment boundaries. We next identify prepositions and nouns of interest using a combination of the Stanford parser and the list of prepositions from Wikipedia [25]. For precise details on how the lexicon is determined, see Section VI-B.

Identifying the arguments to prepositions is done with a small number of rules. The first argument of a preposition may be either a path variable or a floorplan variable corresponding to a noun within the same temporal segment and preceding the preposition. If neither preceded by a conjunction, nor a comma, or if preceded by a WH-determiner, but not a conjunction, the first argument is considered to be the immediately preceding noun. If the preposition is immediately preceded by a conjunction and there are no preceding WH-determiners in the temporal segment, then the first argument is considered to be the path variable. If the preposition is immediately preceded by a conjunction followed by a WH-determiner (e.g., and which) or is immediately preceded by a conjunction and there is no WH-determiner preceding the preposition in the temporal segment, then the first argument is considered to be the noun prior to the immediately preceding noun. There are cases where attachment ambiguity exists. In such cases, the first argument is assumed to be the path variable. The second argument to a preposition is considered to be the noun immediately following the preposition. If there is a temporal break or another preposition between a preposition and the nearest following noun, the preposition is ignored.
Fig. 4. Illustration of the sequence of graphical models induced by a sentence. The sentence is broken into sequential segments, and a path variable (P1, P2) is created for each segment. Next, a floorplan variable (O1–O6) is created for each noun in each segment, applying the noun’s label distribution (in blue) to the variable’s set of labels. Finally, the arguments of each preposition are found, and each preposition’s distributions (in green) over relative positions and velocities are applied between its arguments.

Fig. 5. Illustration of the score function induced by the sentence-segment graphical model from the right side of Fig. 4, using the word models obtained from the learning process. The graphical model is marginalized over all possible mappings from floorplan variables to objects, yielding a scoring function over the position and velocity of the path variable. The velocity is computed as the difference between the position at two adjacent time steps, so that, given the position of the robot at the previous time step, the function can be plotted at each point in space. The score function corresponding to the graphical model is plotted for two different previous positions: the point (3.0, -0.55) (left) and the point (3.0, -2.0) (right). Note that the differing positions for the previous position drastically change the function. In the image on the right, the function prefers points directly between the cone and the previous position, thus satisfying the requirement, which are also to the right of the bottom-most chair. In the left image, the previous position is such that there is no point both between it and the cone and directly to the right of the chair. The optimal point is therefore to move toward the cone, biased somewhat to the right, thus partially satisfying right of the chair. This point has a much lower score that the optimal point on the right plot. Also note that the scoring function correctly prefers points to the right of the chair described in the phrase, and not the other chair or other objects. This is because those mappings of floorplan variable O5 to other objects have a score close to zero at all points. The noun distribution associated with O5 results in low score when the label of the object mapped to O5 is not CHAIR, and the distribution induced by the phrase right of the table, results in low score for any mapping for which the object mapped to O5 is not to the right of O6, and for any mapping for which the label of the object mapped to O6 is not TABLE. Therefore, with the learned word models, only those mappings of floorplan variables to the proper objects significantly influence the score.
Once the arguments to each preposition in a temporal segment have been found, the graphical model is formed as a product of the factors associated with each of the nouns and prepositions. Given a floorplan specifying the positions and class labels of objects, along with the distributions for each noun and preposition, each such graphical model represents the probability that a point in 2D space satisfies the semantics of the corresponding temporal segment of the sentential description. Fig. 5 illustrates the distribution over points in 2D space induced by such a graphical model.

B. Representation of the lexicon

The lexicon specifies the meanings of the nouns and prepositions as a set of probability distributions. The nouns are represented as discrete distributions over the set of class labels. These labels are abstract symbols corresponding to object classes, such as might be obtained by grouping object detections according to class with a clustering algorithm on sensor data. In this work, we do not perform such detection and clustering, but provide such data in the form of a floorplan. For a given floorplan, the robot is provided a list of objects, and clustering, but provide such data in the form of a floorplan. Sensor data. In this work, we do not perform such detection

of the target object (Fig. 6 left). The second, the velocity angle, is the angle between the velocity vector at a waypoint and a vector from the coordinates of the waypoint to the coordinates of the reference object (Fig. 6 right). This second angle is only used for adverbal uses, because it requires computation of the direction of motion, which is determined from temporally adjacent waypoints, and is undefined for stationary objects. This angle is thus taken from the frame of reference of the robot. The von Mises distribution $f(x|\mu, \kappa)$ is given by

$$f(x|\mu, \kappa) = \frac{e^{\kappa \cos(x-\mu)}}{2\pi I_0(\kappa)},$$

where $I_0$ is the modified Bessel function of order 0.

Fig. 1 (bottom left) illustrates how this framework is used to represent the meanings of prepositions. We render the angular distributions as potential fields around the reference object at the center for the position angle, and the target object at the center for the velocity angle. The intensity of a point reflects its probability density. Note that in the idealized distributions of Fig. 1, the distributions are uniform in velocity angle for left of, right of, in front of, and behind and in position angle for towards and away from.

When the $i$th preposition in the lexicon is applied between two variables, whose physical relationship is specified by the position angle $\theta$ and velocity angle $\gamma$ between them, its score $s_{ij}$ is given by

$$s_{i}(\theta, \gamma) = \frac{e^{\kappa_{i,1}\cos(\theta-\mu_{i,1})}}{2\pi I_0(\kappa_{i,1})} - \frac{e^{\kappa_{i,2}\cos(\gamma-\mu_{i,2})}}{2\pi I_0(\kappa_{i,2})},$$

where $\mu_{i,1}$ and $\kappa_{i,1}$ are the location and concentration parameters of the position angle distribution of the $i$th preposition, and $\mu_{i,2}$ and $\kappa_{i,2}$ are the location and concentration parameters of the velocity angle distribution.

C. Computing the graphical model score

Once constructed from a sentence segment, each graphical model induces a distribution over the path variable $p = (p^x, p^y, p^v, p^\gamma)$, conditioned on the $O$ objects in the floorplan $f = (o_1, o_2, \ldots, o_O)$ and the mapping $m$ from the $N$ floorplan variables to floorplan objects. Each element of the mapping $m_n$ is the index of the floorplan object mapped to floorplan variable $n$. Let $a$ be $\{p, o_{m_1}, \ldots, o_{m_N}\}$, a set consisting of the path variable and the floorplan objects mapped to each of the $N$ floorplan variables. Further, let $b_{c,1}$ and $b_{c,2}$ be the indices in $a$ of the target and referent, respectively, of the $c$th

---

1 Without loss of generality, position angles are measured in the frame of reference of the robot at time zero, which is taken to be the origin.
preposition in the graphical model. The 2D world position of the target and referent of the \( c \)th preposition can then be referenced with \((a^y_{b_{c1}}, a^x_{b_{c1}})\) and \((a^y_{b_{c2}}, a^x_{b_{c2}})\), respectively. The velocity vector of the target can similarly be referenced with \((a^y_{b_{c1}}, a^x_{b_{c1}})\). Therefore, the position angle of the target and referent of the \( c \)th preposition in the graphical model is given by

\[
\theta_c = \tan^{-1} \frac{a^y_{b_{c1}} - a^y_{b_{c2}}}{a^x_{b_{c1}} - a^x_{b_{c2}}}
\]

and the velocity angle \( \gamma_c \) between them is given by

\[
\gamma_c = \tan^{-1} \frac{a^y_{b_{c1}} - a^y_{b_{c2}}}{a^x_{b_{c1}} - a^x_{b_{c2}}}
\]

A sentence-segment graphical model’s conditional probability \( P(p|m, f, \Lambda) \) of the path variable given an object mapping \( m \), floorplan \( f \), and lexicon parameters \( \Lambda \) is therefore given by the product of preposition and noun scores:

\[
P(p|m, f, \Lambda) = \prod_{c=1}^{C} d_c(\theta_c, \gamma_c) \prod_{n=1}^{N} w_{en, l_m n}
\]

where \( c \) indexes into the \( C \) prepositions in the graphical model, \( d_c \) is the index in the lexicon of the \( c \)th preposition in the graphical model, \( n \) indexes into the \( N \) nouns in the graphical model, \( e_n \) is the index in the lexicon of the \( n \)th noun in the graphical model, and \( l_m n \) is the class label of the object mapped to the \( n \)th noun.

V. Tasks

We formulate sentential semantics as a variety of relationships between a sentence \( s \), or more precisely its corresponding sequence of graphical models, a path \( p \), which is a sequence of path waypoints, a floorplan \( f \), which is a set of labeled points, and a lexicon \( \Lambda \), which is the collective \( \mu \) and \( \kappa \) parameters for the angular distributions for each of the prepositions and the discrete distributions for each of the nouns. This allows us to accomplish the following tasks:

- **acquisition** Learn a lexicon \( \Lambda \) from a collection of observed paths \( p_t \) taken by the robot in the corresponding floorplans \( f_t \), as described by human-generated sentences \( s_t \).
- **generation** Generate a sentence \( s \) that describes an observed path \( p \) taken by the robot in a given floorplan \( f \) with a known lexicon \( \Lambda \).
- **comprehension** Generate a path \( p \) to be taken by the robot that satisfies a given human-generated sentence \( s \) issued as a command in a given floorplan \( f \) with a known lexicon \( \Lambda \).

A. Acquisition

To perform acquisition, we formulate a large set of hidden Markov models (HMMs), one for each path-sentence pair in the training corpus. Each such HMM has a state \( k \) corresponding to every temporal segment its corresponding training sentence. The observations for each such HMM consist of the sequence of path waypoints in the path-sentence pair. The output model \( R_k \) for each state is the graphical model constructed from that temporal segment, given the current estimate of the parameters in \( \Lambda \) and marginalized over all mappings \( m \) between floorplan variables.

\[
R_k(p_t, f, \Lambda) = \sum_{m} P_k(p|m, f, \Lambda)
\]

The transition matrix for each HMM is constructed to allow each state only to self loop or to transition to the state for the next temporal segment in the training sentence. The HMM is constrained to start in the state associated with the first temporal segment in the sentence associated with each path. Dummy states, with a small fixed output probability, are placed between the states for each pair of adjacent temporal segments, as well as at the beginning and end of each sentence, to allow for portions of the path that are not described in the associated sentence. Fig. 7 illustrates the automatic construction of such an HMM from a sentence.

The HMMs are used to infer the alignment between the densely sampled points in each path and the sequence of temporal segments its corresponding sentence. This process is illustrated in Fig. 8.

The output models for the HMMs are all parametrized by the word meanings from the lexicon \( \Lambda \). Thus, the meaning of each word is constrained by many path-sentence pairs. As illustrated in Fig. 9, this can be thought of as a large (soft) constraint-satisfaction problem. This mutual constraint allows the learning system to gradually infer the unknown mappings between points in the paths and the segments of sentences, and between nouns in the sentences and objects in the floorplans, while simultaneously learning the parameters of the lexicon. Thus, it uses its current estimate of the word meanings to infer which physical relationships between the robot and the objects, or between several objects, are being described, and uses this knowledge to further update the word meanings in order to match the described relationships.

This learning is accomplished by maximizing the summation of the log likelihoods of all HMMs on their corresponding paths through Baum-Welch [27]–[29]. This trains the distributions for the words in the lexicon \( \Lambda \) as they are tied as components of the output models. Specifically, it infers the latent alignment between the large number of noisy robot path waypoints and the smaller number of temporal segments in the training descriptions while simultaneously updating the meanings of the words to match the relationships between waypoints described in the corpus. In this way, the meanings of both the nouns and the prepositions are learned. Fig. 10 illustrates the gradual learning process by showing how the scoring function corresponding to an example phrase begins in a completely meaningless state, but gradually changes to represent the meaning of that phrase as the meanings of the words are gradually learned.

B. Generation

Language generation takes as input a path \( p \) obtained by odometry during human teleoperation of the robot. This path consists of a collection of 2D floor positions sampled at 50Hz.
Fig. 7. A hidden Markov model is created representing the semantics of a sentence. The sentence is broken into segments, and a graphical model is created representing each segment. When a segment cannot be understood, it is pruned, and no graphical model is created. Next, an HMM state is created for each remaining segment. The output model of each such state represents the distribution over the possible positions and velocities of the robot at a given point in time. These output distributions are the graphical models associated with each segment, marginalized over the possible labelings of the floorplan variables. Additional dummy states with uniform output distributions are added at the beginning, end, and between each state. These dummy states allow the HMM to match paths for which the semantics of the sentence are true, but for which there are points in time where the robot does not fulfill any of the stated conditions. The HMM transition distribution encodes the sequence of the sentence by forcing each state to self-transition or pass to the next state, as well as by requiring that the model begin in the first state and end in the last.

Fig. 11. Illustration of the generation algorithm. A disambiguating noun phrase is generated for each floorplan waypoint. Path waypoints are described by prepositional phrases, and then sets of identical phrases are merged into intervals, which are combined to form the sentence.

To generate a sentence, one must select a subsequence of this dense sequence worthy of description.

During generation, we care about three properties: “correctness,” that the sentence be logically true of the path, “completeness,” that the sentence differentiate the intended path from all other possible paths, and “conciseness,” that the sentence be the shortest that does so. We attempt to find a balance between these properties with the following heuristic algorithm (Fig. 11). First, we sample path waypoints in a way that the sampled points evenly distribute along the path. To this end, we downsample the path by computing the integral distance traveled from the initial position for each point in the dense path and selecting a subsequence whose points are separated by 5cm of integral path length. We then produce a prepositional phrase to describe each path waypoint by selecting that preposition with maximum posterior probability with the path waypoint as its first argument and with a floorplan waypoint as its second argument. Identical such choices for consecutive sets of waypoints in the path are coalesced and short intervals of such path prepositional phrases are discarded. We then generate a noun phrase for the object of each path waypoint preposition that refers to that referenced floorplan object. For each floorplan object, we take that noun with maximum posterior probability given the class of the floorplan object. Similarly, for each pair of floorplan objects, we take that preposition with maximum posterior probability to be true of that pair and all other prepositions applied to that pair to be false. Thus when the floorplan contains a single instance of a class, it can be referred to with a simple noun. But when there are multiple instances of a class, the shortest possible noun phrase, with one or more prepositional phrases, is generated to disambiguate.

More formally, let $q(o)$ be the most probable noun for floorplan object $o$ given $\Lambda$. For each pair of floorplan objects $(o, o')$, there exists only one preposition $\phi$ that is true of this pair. Let $u(o)$ be the noun phrase we want to generate to disambiguate the floorplan object $o$ from others $o'$. Then $o$ can be referred to with $u(o)$ unambiguously if (a) $u(o) = (q(o), \{\})$ is unique; or (b), there exists a collection of prepositional phrases $\{\phi(o, o')\}$ such that formula $u(o) = (q(o), \{(\phi, u(o'))\})$ is unique. To
The robot started to the right of the stool, traveled toward the bag, doubled back to pass to the right of the bag, stool, then chair and slowly circled in front of the cone, ending just to the right of it.

Fig. 8. Illustration of aligning an example sentence-path pair from the training set. An HMM is produced to represent the semantics of the sentence (top left). Given a floorplan and path (top right), the HMM is used during learning to perform an alignment between the states (and therefore the temporal segments of the sentence) and the densely sampled waypoints of the path. Each HMM output model computes the score of the position and velocity at each path waypoint (middle row). Because the preposition and noun distributions are unknown at the start of the learning process, the labels of the floorplan variables, which map nouns in the sentence to objects in the floorplans, are also unknown. Therefore each state’s output score is the likelihood of the associated graphical model, marginalized over all possible mappings of floorplan variables to labels. These scores, along with the HMM transition model, are used with the forward-backward algorithm to compute the probability of the HMM being in each state (bottom row), along with the HMM likelihood. Prior to learning, all preposition and noun distributions are random. During acquisition of such meanings, the model is updated to increase the overall HMM likelihood summed over all training samples. At each iteration, this concentrates the probability mass of the distributions associated with the prepositions for each HMM-state output model at those angles seen among waypoints and objects at those time steps at which the probability of the HMM being in that state is high. It also concentrates the probability mass of the object-label distributions in those bins associated with the mappings corresponding to high HMM likelihoods. This example shows the scores and HMM state-probability assignments using the word models after the learning process is complete. Note that while this sentence is true of the path, there are portions of the path (such as from approximately time step 40 to time step 55, during which the robot is traveling away from the bag toward the cone) which are not described by the sentence. Thus, portions of the path are automatically assigned to the dummy states, preventing them from interfering with the learning process.

produce a concise sentence, we want the size of the collection of prepositional phrases in step (b) above to be as small as possible. However, finding the smallest collection of modifiers is NP-hard [30]. To avoid exhaustive search, we use a greedy heuristic that biases towards adding the least frequent pairs \((\phi, u(o'))\) into the collection until \(u(o)\) is unique. This results in a tractable polynomial algorithm. After we get \(u(o)\), we map it to a noun phrase by simple realization, for example:

\[(\text{TABLE, \{(LEFTOF, CHAIR), (BEHIND, TABLE)\}})\]

\[\downarrow\]

the table which is left of the chair and behind the table

C. Comprehension

To perform comprehension, we use gradient ascent to optimize a scoring function with respect to an unknown path \(p\)

\[p^* = \arg \max_p R(s, p, f, \Lambda)\]
where $R(s, p, f, \Lambda)$ is the product of the graphical model likelihoods $P_k(p_k|m, f, \Lambda)$ from (1) constructed from the temporal segments of the sentence $s$. The unknown path $p$ is constructed to contain one path waypoint $p_k$ for each temporal segment $k$ in the sentence, whose locations are optimized to maximize the scoring function, and thus maximize the degree to which these waypoints satisfy the semantics of the sentence.

The above optimization computes a MAP estimate of the product of the likelihoods of the graphical models associated with the sentence $s$. These graphical models represent the semantics of the sentence, but do not take into account constraints of the world, such as the need to avoid collision with the objects in the floorplan. Further, the scoring function as stated can be difficult to optimize because the velocity angle computed between two waypoints becomes increasingly sensitive to small changes in their positions as they become close together. To remedy the problems of the path waypoints getting too close to objects and to each other, additional factors are added to the graphical models. A barrier penalty $B(r)$ is added between each pair of a path waypoint and floorplan waypoint as well as between pairs of temporally adjacent path waypoints to prevent them from becoming too close. We use the formula

$$B(r) = \text{SMOOTHMAX} \left( 1, 1 + \frac{2r_1 + r_2}{r} \right)^{-1}$$

where $r$ is the distance either between a path waypoint and an object or between two path waypoints, and where $r_1$ and $r_2$ are the radii of the two things being kept apart, either the robot or an object. This barrier is approximately 1 until the distance between the two waypoints becomes small, at which point it decreases rapidly, pushing them away from each other by approximately the robot radius. For the penalty between the path waypoints and objects, meant to prevent collision, both the robot radius and object radii are assumed to be 40cm. For the penalty between temporally adjacent path waypoints, meant to ease the optimization problem, $r_1$ and $r_2$ are set to 10cm. Finally, because our formulation of the semantics of prepositions is based on angles but not distance, there is a large subspace of the floor that leads to equal probability of satisfying each graphical-model factor, i.e., the cones in Fig. 1. This allows a path to satisfy a prepositional phrase like to the left of the chair while being very far away from the chair, which, while technically correct, can result in paths
which appear to a human to be infelicitous. To remedy this, we encode a slight preference for shorter distances by adding a small attraction $A(r) = \exp(-r/100)$ between each path waypoint and the floorplan waypoints selected as its reference objects, where $r$ is the distance between the path waypoint and the target object of a preposition. The score optimized is the product of the graphical-model factors for each path waypoint along with the barrier and attraction terms. An example of the scoring function corresponding to the example phrase *toward the chair left of the bag*, together with the additional terms, is shown in Fig. 13. Its gradient with respect to the path waypoint locations is computed with Automatic Differentiation [31]. The sequence of path waypoints maximizing this product is then found with gradient ascent. The individual points cannot be optimized independently because each graphical model score depends on the velocity, and thus the previous point. The score is optimized repeatedly with subsets of the waypoints increasing in size. The waypoints are added sequentially, each time initializing the newest point 10cm from the last point in such a way that direction of motion is maintained. Then, the product of scores corresponding to the current set of points is optimized. In the final stage of optimization, all points have been added, and the entirety of the score is optimized. This process helps to prevent the optimization procedure from becoming stuck in local optima.

Fig. 12 shows the effect of small differences in the input
The robot went away from the cone then went left of the box which is left of the chair and behind the cone then went towards the stool.

The robot went away from the cone then went right of the box which is left of the chair and behind the cone then went towards the stool.

The robot went away from the cone then went behind the box which is left of the chair and behind the cone then went towards the stool.

The robot went away from the cone then went behind the box which is right of the chair and which is behind the cone then went towards the stool.

The robot went away from the cone then went behind the box which is right of the chair and which is behind the cone then went towards the stool.

The robot went away from the cone then went behind the box which is left of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is right of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is left of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is right of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is left of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is right of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is left of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is right of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is left of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is right of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is left of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is right of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is left of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is right of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is left of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is right of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is left of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is right of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is left of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is right of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is left of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is right of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is left of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is right of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is left of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is right of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is left of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is right of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is left of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is right of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is left of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is right of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is left of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is right of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is left of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is right of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is left of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is right of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is left of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is right of the chair and in front of the cone then went towards the stool.

The robot went away from the cone then went behind the box which is left of the chair and in front of the cone then went towards the stool.

The barrier penalties prevent the path waypoints from being chosen close to objects, but do not prevent the paths between them from doing so. Therefore, a postprocessing step performs obstacle avoidance by adding additional path waypoints as needed to ensure that the straight-line path between any two adjacent path waypoints does not pass too close to any floorplan object. This is done by looping over all line segments between temporally adjacent waypoints, checking whether the distance between each object and that point \( v \) on the line closest to that object is less than the sum of the object and robot radii. If so, a new waypoint is created at the point closest to \( v \) which is separated from the object by the sum of the two radii. This process is then repeated recursively on all modified line segments until no segment passes too close to any object.

VI. Experiments

We conducted an experiment as outlined in Fig. 1. A corpus of robot paths paired with sentences describing those paths was first collected and used to learn a lexicon. This lexicon was then used to automatically generate sentences from a new set of paths and to produce and follow paths satisfying a new set of sentences. Fig. 14 illustrates examples of input to the learning system, the learned lexicon, and examples of the input to and output from the generation and comprehension systems.

All sentences were obtained from independent, anonymous workers on AMT. All paths were recorded using odometry and sensor data from the robot. A second set of judgments were obtained from AMT workers in which they judged the degree to which the sentences and paths matched. Such judgments were obtained for the robot paths automatically driven according to human sentences, for automatically produced sentences describing human-driven paths, and also for the sentences produced by AMT workers to describe human-driven paths. This allows us to compare the performance of the automatic systems against that of AMT workers. Fig. 15 depicts the performance of both AMT workers and of our system.

A. Dataset collection

Three sets of robot paths and sentences were collected. The first, called the acquisition corpus, consisted of 750 path-sentence pairs, 3 sentences for each of 250 robot paths. The second, called the generation corpus, consisted of 100 robot paths. The third, called the comprehension corpus, consisted
The robot went to the right of the table which is to the right of the table, then away from the table which is right of the table, then away from the cone, then toward the table which is right of the table, then toward the stool, then left of the stool.

The robot went by heading towards the stool, turning around when it reached the right side of the stool, then it headed towards the cone, turning around in front of the cone and turned around and ended in front of the stool and behind the cone.

The robot loops around the left side of the cone and then goes underneath the cone.

The robot went behind the back of the chair, then went in front of the stool but behind the cone, then went away from the stool to the left and behind it.

The robot went behind the cone and then turned around and went further behind the cone to the right of the chair.

The robot went toward and around the chair, right of the box and bag, then in front of the box.

Fig. 14. Example experimental runs, 6 for each of acquisition, generation, and comprehension. Our source code and dataset will be available upon publication.

of 300 path-sentence pairs, 3 sentences for each of 100 robot paths.

All sentences were obtained from anonymous workers on AMT. These workers were asked to provide a sentence describing a robot path as depicted in an image. They were asked to describe the path in terms of its relation to objects in the floorplan, but were not given any restrictions on the syntax of the sentence. Neither were they told what the sentences were for. Therefore the sentences are not artificially mechanical or specific, as they might have been had they known that they were to be used by a robot. Rather, they are representative of human path descriptions in this domain. Each of the paths used to elicit sentences from workers was obtained by a human driving the robot in a path through a randomly generated floorplan in accordance with a randomly generated sentence.

For the acquisition corpus, 10 floorplans were generated, each with 25 random sentences. Each floorplan contained four random objects, with up to one duplicate object. The objects were placed randomly at the corners of floor tiles in a large room. The random sentences used to guide the driving of the
acquisition paths used to elicit sentences contained a sequence of two or three instructions to move according to random prepositions chosen from left, right, front, behind, towards, and away with respect to randomly chosen objects in the floorplan. For the generation and comprehension corpora, the floorplan contained five random objects, also limited to one duplicate, which were placed randomly on the corners, centers, or edge centers of the floor tiles. The randomly generated sentences for these corpora were generated similar to those of the acquisition corpus, only longer, with a sequence of five or six instructions. For each such sentence, the robot was driven in a path according to the instructions of the random sentence. The human driver was allowed to drive freely, so long as the sentence remained true of the path. Therefore, while these paths do contain, in the proper order, portions which depict the described physical relationships, the driven paths are generally more complex than the original randomly generated sentence.

Images of these paths were given to the AMT workers to elicit sentences. Workers were shown paths in this way in order to elicit sentences which describe a variety of paths. Many of these elicited sentences contain ambiguity, misspellings, grammatical errors, or describe relations which are untrue or impossible, such as describing a chair as the chair to the right of the chair, when there is, in fact, only a single chair. We corrected for obvious misspellings, but did not otherwise make modifications to the sentences. Quantitative evaluation of the quality of the sentences from a second round of AMT judgments can be seen in Fig. 15, where the results of the human-generated sentences judged against the paths used to elicit them are shown in dark blue (acquisition sentences) and light blue (comprehension sentences). Only about 40% of human-generated sentences received the highest possible rating from human judges on AMT, with the rest being judged less than 80% correct or complete. Our methods are robust enough to handle such errors gracefully: the learning system learns the correct meaning of words despite the noisy and ambiguous training data, and the comprehension system ignores parts of sentences it cannot understand.

B. Experimental evaluation

The path-sentence pairs in the acquisition corpus were used to learn a lexicon of word models. The paths in the generation corpus were used to test the robot’s ability to use its learned lexicon to generate a sentence which describes a given path. The sentences in the comprehension corpus were used to test the robot’s ability to use its learned lexicon to automatically generate and follow a path described by a sentence.

We determined the nouns and prepositions to learn as follows. We identified prepositions of interest as words which appear in the Wikipedia list of prepositions, and which appear more than 100 times in our combined corpus of 1050 sentences, (excluding in, on, and to). This resulted in the following list of prepositions: left, right, front, behind, towards, and away. We identified nouns of interest as words which have not been identified as prepositions, and which have been tagged as nouns by the Stanford parser more than 100 times in our corpus (excluding the word robot). This resulted in the following list of nouns: bag, box, chair, cone, stool, and table.

Among the 750 sentences in the acquisition corpus a number of extremely long sentences which approach the limit of the Stanford parser, and which slow down the learning process. Therefore, the sentences were sorted according to the number of references to objects, and the 600 shortest sentences were used for training. This resulted in the exclusion of sentences with more than 16 object references. The acquisition system used the resulting 600 path-sentence pairs to learn the lexicon of word meanings.

After learning the meanings of the words, the recorded paths from the generation corpus were used to automatically produce sentential descriptions and the sentences from the comprehension corpus were used to automatically drive the robot, recording its path through odometry. A second round of AMT judgments were obtained, judging the degree to which the sentence and path in each pair match. Such judgments were obtained for the robot paths automatically driven according to human sentences, for the automatically produced sentences describing human-driven paths, and also for the sentences produced by AMT workers to describe human-driven sentences. This allowed us to compare the performance of the automatic systems with that of AMT workers through four multiple-
VII. Conclusion

We demonstrate a novel approach for grounding the semantics of natural language in the domain of robot navigation. Sentences describe paths taken by the robot relative to other objects in the environment. The meanings of nouns and prepositions are trained from a corpus of paths driven by a human teleoperator annotated with sentential descriptions. These can then support both automatic generation of sentential descriptions of new paths as well as automatic driving of paths to satisfy navigational goals specified in provided sentences. This is a step towards the ultimate goal of grounded natural language that allows machines to interact with humans when the language refers to actual things and activities in the real world.

Acknowledgments

This research was sponsored, in part, by the Army Research Laboratory, accomplished under Cooperative Agreement Number W911NF-10-2-0060, and by the National Science Foundation under Grant No. 1522954-IIS. The views, opinions, findings, conclusions, and recommendations contained in this document are those of the authors and should not be interpreted as representing the official policies, either express or implied, of the Army Research Laboratory, the National Science Foundation, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation herein.

References


Daniel Paul Barrett received the BScEng degree from Purdue University in 2011. He is currently a Ph.D. student in the School of Electrical and Computer Engineering at Purdue University. His research interests include computer vision, robotics, and artificial intelligence, particularly their intersection, where a robot perceives, learns about, and acts on the world through noisy real-world camera and sensor input.

Scott Alan Bronikowski received the B.S. degree in electrical and computer engineering from the U.S. Military Academy, West Point, NY, in 1999, and the M.S. degree in electrical and computer engineering from Kansas State University in 2009. He served as an instructor and an assistant professor in the Department of Electrical Engineering and Computer Science at the U.S. Military Academy from 2009 to 2012. He is currently a Ph.D. student in the School of Electrical and Computer Engineering at Purdue University. His research interests include robotics, artificial intelligence, natural-language processing, and computer vision, specifically uses of NLP, CV, and AI in general to improve the methods through which humans interact with mobile robots.

Haonan Yu is currently a PhD student in the school of Electrical and Computer Engineering at Purdue University. Before that, he received his B.S. of Computer Science from Peking University, China. His research interests are Computer Vision and Natural-Language Processing. He is the recipient of the Best Paper Award of ACL 2013.

Jeffrey Mark Siskind received the B.A. degree in computer science from the Technion, Israel Institute of Technology in 1979, the S.M. degree in computer science from MIT in 1989, and the Ph.D. degree in computer science from MIT in 1992. He did a postdoctoral fellowship at the University of Pennsylvania Institute for Research in Cognitive Science from 1992 to 1993. He was an assistant professor at the University of Toronto Department of Computer Science from 1993 to 1995, a senior lecturer at the Technion Department of Electrical Engineering in 1996, a visiting assistant professor at the University of Vermont Department of Computer Science and Electrical Engineering from 1996 to 1997, and a research scientist at NEC Research Institute, Inc. from 1997 to 2001. He joined the Purdue University School of Electrical and Computer Engineering in 2002 where he is currently an associate professor.