Driving Under the Influence (of Language)

Daniel Paul Barrett, *Member, IEEE*, Scott Alan Bronikowski, *Member, IEEE*, Haonan Yu, *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*— Cognitive human–robot interaction, learning and adaptive systems, natural language robot interaction and control, wheeled robots.**

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 a novel method to represent the meaning of a sentence that describes a path driven by a robot through an environment containing a number of objects. An example of such a sentence is *The robot went toward the box which is left of the chair and behind the cone and then went in front of the stool.* Such sentences are sequences of descriptions in terms of objects in the environment. Nouns in the descriptions indicate the class of the objects involved, such as *box* or *chair*. However, the nouns do not specify exactly which object in the environment is being referenced, as, for example, there may be more than one *box* in the environment. This introduces the potential for ambiguity. Prepositions in the sentences, such as *in front of* and *left of*, are used to describe the changing position of the robot over time (e.g., *the robot went in front of the stool*), as well as to describe the relative positions of the objects in the environment (e.g., *the box which is left of the chair*). We refer to the former kind of usage as *adverbial* and the latter as *adjectival*. Many prepositions, like *in front of*, can have both adverbial usage, as in *the robot went in front of the chair*, and adjectival usage, as in *the chair in front of the table*. Both adverbial and adjectival usage can be nested to arbitrary depth, as in *toward the chair which is in front of the table which is right of the stool which is …* Both can also be combined with conjunctions to describe a single object in terms of several others, as in *the box which is left of the chair and behind the cone*, or to describe the position of the robot at a particular point in time in terms of multiple objects, as in *went toward the chair and left of the table*. The use of nesting and conjunction allows both rich description of the path of the robot and disambiguation of the specific objects used to describe the robot motion.

We represent the meaning of a sentence through a scoring function that takes as input a sentence, a robot path, a floorplan that specifies the locations of objects in the robot's environment, and a set of parameters defining the meaning of words, and returns high score only when the sentence is true of the path in the given environment. This method allows a single unified representation of word and sentence meaning to be used to perform three tasks. The first is word meaning *acquisition*, in which the meanings of prepositions like *toward* and nouns like *table* are learned from a data set of sentences describing paths driven by a robot. The second is sentence *generation*, in which previously learned words are used to automatically produce a new sentence that correctly describes an input robot-driven path. The third is sentence *comprehension*, in which previously learned words are used to automatically produce a new path that satisfies an input sentential description. These three tasks are all accomplished by optimizing the scoring function relative to different variables: acquisition is accomplished by optimizing the word-meaning parameters, generation is accomplished by optimizing the sentence, and comprehension is accomplished by optimizing the path. Fig. 1 shows a data-flow diagram of our system.

We have conducted experiments with an actual radio-controlled robot that demonstrate all three tasks: acquisition,

Manuscript received January 26, 2016; revised January 4, 2017; accepted April 4, 2017. Date of publication June 9, 2017; date of current version June 21, 2018. This work was supported in part by the Army Research Laboratory accomplished under under Cooperative Agreement W911NF-10-2-0060 and in part by the National Science Foundation under Grant 1522954-IIS. *(Corresponding author: Jeffrey Mark Siskind.)*

D. P. Barrett was with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA. He is now with the Sandia National Laboratories, Albuquerque, NM 87123 USA (e-mail: dpbarret@purdue.edu).

S. A. Bronikowski was with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA. He is now with General Motors, Milford, MI 48380 USA (e-mail: scottbronikowski@gmail.com).

H. Yu was with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA. He is now with Baidu Research, Sunnyvale, CA 94089 USA (e-mail: haonanu@gmail.com).

J. M. Siskind is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: qobi@purdue.edu)

This paper contains one or more figures available online at http://ieeexplore.ieee.org (File size: 19 MB). The code and dataset for this work is available at upplysingaoflun.ecn.purdue.edu/ qobi/duil-dataset.tgz

Digital Object Identifier 10.1109/TNNLS.2017.2693278

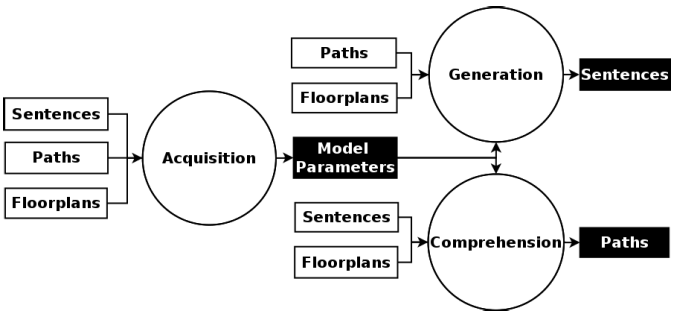


Fig. 1. Data-flow diagram of our system. Boxes with black text on white background denote data generated by humans, while boxes with white text on black background denote machine-generated data.

generation, and comprehension. We demonstrate successful completion of all three tasks on hundreds of driving examples. The sentences employed for the acquisition and comprehension experiments were all elicited from anonymous workers on Amazon Mechanical Turk (AMT).We evaluate the fidelity of the sentential descriptions produced automatically in response to manual driving, i.e., generation, and the fidelity of the driving paths induced automatically to fulfill natural-language commands, i.e., comprehension, by presenting the pairs of sentences together with the associated paths to anonymous human judges on AMT. Thus, we have two distinct rounds of human work on AMT as part of our evaluation: the first round generates sentences used for acquisition and comprehension, while the second round evaluates sentences produced by generation and paths produced by comprehension. These two rounds employ different random workers from the AMT pool. 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%. The machine-generated sentences were judged to be approximately 12% less concise than human-generated sentences.

II. Overview

Our scoring function $\mathcal{R}(\mathbf{s}, \mathbf{p}, \mathbf{f}, \Lambda)$ represents the truthfulness of a sentence $\mathbf{s}$ relative to path $\mathbf{p}$ driven by the robot in an environment described by a floorplan $\mathbf{f}$, given a lexicon Λ of word meanings. This function returns a high score when the sentence is true of the path taken by the robot through the environment, and a low score if it is false. A path $\mathbf{p}$ is represented as a sequence of 2D robot positions over time. A floorplan $\mathbf{f}$ consists of a set of 2D object positions with associated class labels, such as there being a *chair* at offset (2.5 m north, 1.7 m east) from the origin. For example, the sentence *The robot went toward the chair which is behind the table and then went away from the stool* makes a sequence of assertions about the position and velocity of the robot relative to two objects, the *chair* and the *stool*. It also makes an assertion about the relative positions of two of the objects, the *chair* and the *table*. A sentence $\mathbf{s}$ will have a certain degree of truthfulness describing a path $\mathbf{p}$ in a floorplan $\mathbf{f}$. This truthfulness depends upon the relative position and velocity of the robot at different points in time with respect to the positions of the objects, as well as the relative positions of the objects with respect to each other. Since a sentence, or sequence of sentences, describing a path can make a sequence of assertions, computing the degree of truthfulness requires performing a temporal alignment between the elements in this sequence of assertions and portions of the robot path. A sentence may be true even if there are portions of the path that are not described. A sentence may be false, even if all the elements in the sequence of assertions have a corresponding portion of the path for which they are true, if they do not occur in the correct order. Thus, the scoring function must find the maximally true alignment between each such part of the sentence and a portion of the path such that the ordering of the path portions matches the ordering of the sentence parts and each part of the sentence is maximally true of its corresponding path portion.

The scoring function $\mathcal{R}(\mathbf{s}, \mathbf{p}, \mathbf{f}, \Lambda)$ is compositional: the truthfulness of a sentence $\mathbf{s}$ is determined by evaluating and combining word-specific scoring functions which represent the meanings of nouns and prepositions in $\mathbf{s}$, such as *chair* and *toward*. The meaning of each word is represented through a probability distribution whose specific form is determined by a set of word-specific parameters, Λ. Compositionality means that the aggregate scoring process for two different sentences may differ yet share parameters for the words in common. This allows our method to be *generative*: a combinatorially large set of possible sentences can be supported with even a small lexicon of nouns and prepositions. Moreover, this lexicon can be learned with a relatively small set of training examples.

This method makes possible three different use cases, simply by optimizing the function $\mathcal{R}$ with respect to different arguments. Automatic word-meaning acquisition is possible by optimizing $\mathcal{R}(\mathbf{s}, \mathbf{p}, \mathbf{f}, \Lambda)$ with respect to the latent parameters Λ to maximize the predicted truthfulness of a data set of sentences $\mathbf{s}_l$ describing robot paths $\mathbf{p}_l$ through floorplans $\mathbf{f}_l$

$$\hat{\Lambda} = \arg \max_{\Lambda} \prod_{l} \mathcal{R}(\mathbf{s}_l, \mathbf{p}_l, \mathbf{f}_l, \Lambda). \tag{1}$$

The learned word meanings Λ can then be used to perform two other tasks. Automatic generation of a sentence $\mathbf{s}$ that describes a robot path $\mathbf{p}$ through a floorplan $\mathbf{f}$ is possible by optimizing $\mathcal{R}(\mathbf{s}, \mathbf{p}, \mathbf{f}, \Lambda)$ with respect to the sentence $\mathbf{s}$ to maximize its truthfulness given the path $\mathbf{p}$, floorplan $\mathbf{f}$, and model parameters Λ

$$\hat{\mathbf{s}} = \arg \max_{\mathbf{s}} \mathcal{R}(\mathbf{s}, \mathbf{p}, \mathbf{f}, \Lambda). \tag{2}$$

Automatic production of a robot path $\mathbf{p}$ that satisfies a given sentence $\mathbf{s}$ (thus automatic comprehension of its meaning) is possible by optimizing $\mathcal{R}(\mathbf{s}, \mathbf{p}, \mathbf{f}, \Lambda)$ with respect to the positions in the path $\mathbf{p}$ to maximize the truthfulness of the given sentence $\mathbf{s}$ in relation to the path $\mathbf{p}$, floorplan $\mathbf{f}$, and model parameters Λ

$$\hat{\mathbf{p}} = \arg \max_{\mathbf{p}} \mathcal{R}(\mathbf{s}, \mathbf{p}, \mathbf{f}, \Lambda). \tag{3}$$

Acquisition is difficult because of both the natural ambiguity of the sentences, and ambiguity resulting from the fact that the meanings of the words are not known beforehand. A sentence does not specify which portion of each path is described by each of its parts. The alignment process inside the scoring function must determine this. Further, the sentences do not necessarily specify the particular objects being described or referenced, because nouns specify only the object classes, not specific objects in the floorplan. However, without knowing the meanings of the nouns, even the classes of the referenced objects are unknown. A sentence may include prepositional phrases to disambiguate the referenced objects, but this does not provide the information to determine the referenced objects during early stages of learning when the word meanings are still unknown. A single path–sentence pair has too much ambiguity to determine which objects are being referenced, or which parts of the sentence correspond to each portion of the path, let alone what relationships among the objects, or between the path and the objects, correspond to the meanings of the prepositions to be learned. However, the shared parameters between different sentences that arise from the use of some of the same words to describe different situations make it possible to use a number of path–sentence pairs together to disambiguate the sentence meanings and learn the word meanings through a gradual iterative learning procedure.

Generation is difficult for two reasons. First, the input path $\mathbf{p}$ is a dense sequence of points which must be automatically segmented into portions, each of which is to be described by part of the sentence $\mathbf{s}$. Second, the generated sentential parts must be unambiguous and concise. We wish to generate a sentence that is true of the path, a sentence which is true only of that path and of no other qualitatively different paths, and the shortest sentence for which this is the case. This sentence may wish to situate the path relative to specific objects in the floorplan. There can be more than one instance of a given object class in the floorplan, so a complex noun phrase must be generated to uniquely refer to that object. We wish to find the shortest noun phrase that does so.

Comprehension is difficult because the input sentence $\mathbf{s}$ is not a complete specification of the desired robot path $\mathbf{p}$; it only incompletely specifies constraints over $\mathbf{p}$. Path planning must be performed to find a complete path specification that not only satisfies the sentential constraints but also avoids obstacles.

III. Representing the Meaning of a Sentence

The meaning of a sentence or sequence of sentences can be captured by representing the assertions they make. A sentence describing a robot path with respect to the environment makes assertions about the robot path and the objects in a floorplan. In order for the sentence to be true, all the assertions must also be true. For example, the sentence *The robot went toward the chair behind the table, and then went in front of the stool*, denotes a sequence of two sets of assertions. The first set consists of four assertions: 1) the robot's path brings it *toward* an object; 2) that object is called a *chair*; 3) the *chair* is *behind* another object; and 4) that other object is called a *table*. The second set consists of two assertions: 1) the robot's path brings it in *front* of an object and 2) that object is called a *stool*. Further, the sequential nature of the sentence provides an additional assertion that the second set of assertions must be fulfilled after the first is fulfilled in order for the sentence to be true.

We represent the meaning of such a sequence of assertions with a sequence of graphical models which are grounded in a *path*, which is a sequence of *waypoints*, the 2D positions of the robot over time, and a *floorplan*, which consists of a set of *floorplan objects*, labeled 2D points representing the position and class of objects in the environment. Each graphical model is a product of factors. Each factor is a probability distribution representing one of the assertions in the sentence, and corresponds to a word or clause in a sentence. Continuing the above example, there is a factor representing the assertion that the robot's path moves *toward* an object. This factor is a probability distribution between a *path variable*, which is a pair of 2-D vectors representing the position and velocity of the robot at a particular time, and a *floorplan variable*, which is a labeled 2-D Cartesian coordinate representing the class and position of a floorplan object. The model parameters Λ define the shape of each factor distribution and thus define the meaning of each word, such as *toward*. Such a distribution encodes the meaning of a preposition like *toward* by placing probability mass on certain relative velocities and positions between the path variable and the floorplan variable that satisfy the spatial relationship defined by the word. In general, a factor distribution corresponding to a preposition can be applied between a path variable and a floorplan variable to define its adverbial usage, as shown above, or between two floorplan variables, to define its adjectival usage, such as the distribution corresponding to one object being *behind* another. Other distributions can encode the meanings of nouns like *chair* or *table* by placing probability mass on certain values of a floorplan variable's label. The exact details of how each factor distribution is defined in terms of the parameters in Λ are described in Section III-B.

The product of factors in each graphical model captures the meaning of each set of assertions in the sequence. Just as the sentence is false if any individual assertion is false, the product of factors is close to zero if any of the individual factors is close to zero. Given any assignment of values to the path variable and floorplan variables, the graphical model will produce a score value corresponding to the veracity of that sentence clause with respect to the robot path and objects defined by those path and floorplan variables. The meaning of a sentence or sequence of sentences is therefore captured by a corresponding sequence of graphical models when they are constrained to be satisfied in the proper sequence.

A. Constructing Graphical Models From a Sentence

We automatically generate such a sequence of graphical models directly from a sentence or sequence of sentences. The sentence(s) are first broken into temporal segments using a subset of the rules of English grammar and a graphical model is produced for each segment. Fig. 2 illustrates this process, showing an example sentence and the sequence of

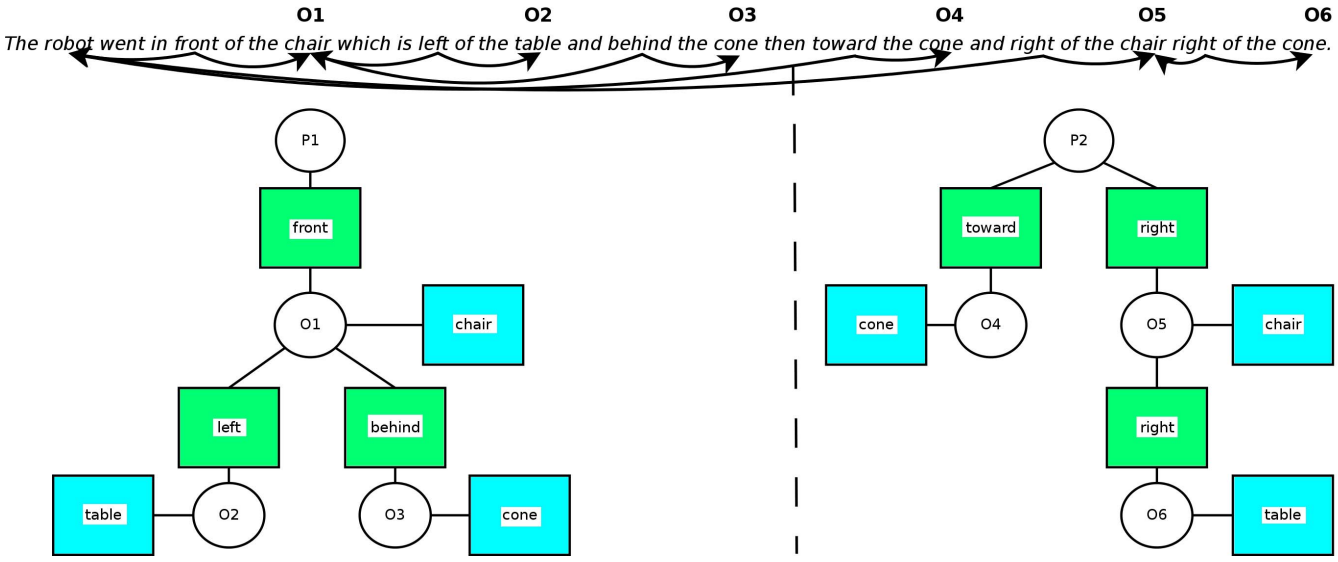


Fig. 2. Illustration of the sequence of graphical models induced by a sentence. The sentence is broken into sequential segments, and a path variable (P1 and P2) is created for each segment. Next, a floorplan variable (O1–O6) is created for each noun in each segment, applying the noun's label distribution (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.

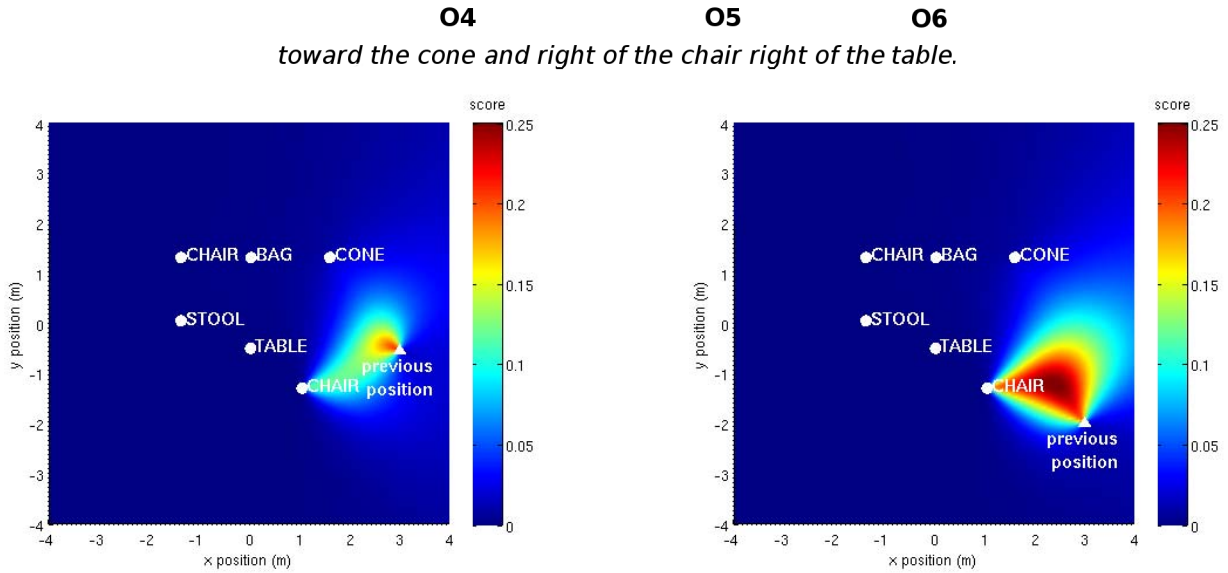


Fig. 3. Illustration of the score function induced by the sentence-segment graphical model from Fig. 2 (right), using the word models obtained from the learning process. Moving from the previous robot position to a high scoring (red) region satisfies the semantics of the phrase. Note that the differing positions for the previous position drastically change the function.

two graphical models produced from it. A path variable (white circles labeled P1 and P2 in Fig. 2) is created for each graphical model, representing the position and velocity of the robot during that segment of the path. A floorplan variable (white circles labeled O1–O6) is created for each noun instance in the sentence(s), representing the position and label of an object in the floorplan. The position and label of a floorplan variable can be taken from the 2D position and class label of any object in the provided floorplan. Each noun in a sentence also results in a univariate distribution (blue rectangles connected to the associated floorplan variable) over possible class labels that the floorplan variable may take.

Each preposition in a sentence induces a joint distribution between the two variables to which it is applied. These are the target and referent objects. Fig. 2 shows the distribution for each preposition as a green rectangle connecting the target object (top circle) and referent object (bottom circle). For example, in the clause *The robot went in front of the chair which is left of the table and behind the cone*, the phrase *the*

robot went in front of the chair results in the *front* preposition distribution applied between the target object, path variable P1 (the robot), and the referent object, floorplan variable O1 (the chair). There is a noun (*chair*) distribution also applied to the label of O1, along with an additional two other preposition distributions (*left* and *behind*) resulting from the other phrases, *left of the table* and *behind the cone*. Note that O1 is the referent object of *front* but the target object of *left* and *behind*.

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. For a given assignment of values to each path variable (position and velocity) and floorplan variable (position and label), the graphical model's probability represents the degree to which those values satisfy the meaning of the sentence.

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. For example, objects of class **bag** might have class label CLASS0, while objects of class **stool** might have label CLASS4. These come from the provided floorplans, which are lists of objects each consisting of a 2D position and class label. Observe that the class labels do not uniquely specify an object in a floorplan because there are often multiple objects of the same class in a given floorplan.

Each noun i in the lexicon Λ consists of a set of weights w_{ij} which score the mappings between it and each possible label j. When a noun distribution is applied to a floorplan variable, it gives a score to the label assigned to that variable.

Each floorplan variable generated from a sentence can be mapped to one of the objects in a floorplan, taking its position and class label. When mapped to the kth object, whose label is l_k and which resides at location (x_k, y_k), the score of the noun distribution i applied to that variable is w_{i,l_k}.

Prepositions specify relations between target objects and referent objects. The target object of a preposition may be an object in the floorplan when the preposition is used adjectivally to describe a noun or may be a waypoint in the robot's path when used adverbially to describe the robot's motion. For example, in *the chair to the left of the table*, the floorplan variable corresponding to the noun *chair* is the target object and the floorplan variable corresponding to *table* is the referent object, whereas in the phrase, *went toward the table*, the path variable is the target object while the floorplan variable corresponding to *table* is the referent object. The lexical entry for each preposition in Λ is specified as the location μ and concentration κ parameters for two independent von Mises distributions [1] over angles between target and referent objects. One, the *position angle*, is the orientation of a vector from the coordinates of the referent object to the coordinates of the target object [Fig. 4 (left)].[1] The second, the *velocity angle*, is the angle between the velocity vector of the target object and a vector from the coordinates of the target object to the coordinates of the referent object [Fig. 4 (right)]. This second angle is only used for adverbial uses describing the robot path, because it requires computation of the direction of motion, which is undefined for stationary objects. This angle is thus taken from the frame of reference of the robot.

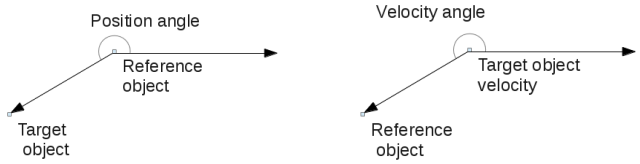


Fig. 4. How position angles (left) and velocity angles (right) are measured.

The von Mises distribution defining each angular distribution $\nu(\alpha|\mu, \kappa)$ is given by

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

where I_0 is the modified Bessel function of order 0.

When the ith preposition in the lexicon is applied between two variables, whose physical relationship is specified by the position angle θ and velocity angle γ between them, its score z_i is given by

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

where $\mu_{i,1}$ and $\kappa_{i,1}$ are the location and concentration parameters of the position angle distribution of the ith 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 $\rho = (\rho^x, \rho^y, \rho^{v_x}, \rho^{v_y})$, conditioned on the K objects in the floorplan $\mathbf{f} = (o_1, \ldots, o_K)$ and the latent 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. This latent mapping designates which objects in the floorplan are referred to by each noun in the sentence. Let a be $\{\rho, o_{m_1}, \ldots, o_{m_N}\}$, 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 cth preposition in the graphical model. The 2-D world position of the target and referent of the cth preposition can then be referenced with $(a^x_{b_{c,1}}, a^y_{b_{c,1}})$ and $(a^x_{b_{c,2}}, a^y_{b_{c,2}})$, respectively. The velocity vector of the target can similarly be referenced with $(a^{v_x}_{b_{c,1}}, a^{v_y}_{b_{c,1}})$. Therefore, the position angle of the target

[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.

and referent of the cth preposition in the graphical model is given by

$$\theta_c = \tan^{-1} \frac{a^y_{b_{c,1}} - a^y_{b_{c,2}}}{a^x_{b_{c,1}} - a^x_{b_{c,2}}}$$

and the velocity angle γ_c between them is given by

$$\gamma_c = \tan^{-1} \frac{a^{v_y}_{b_{c,1}}}{a^{v_x}_{b_{c,1}}} - \tan^{-1} \frac{a^y_{b_{c,2}} - a^y_{b_{c,1}}}{a^x_{b_{c,2}} - a^x_{b_{c,1}}}.$$

A sentence-segment graphical model's conditional probability $\psi(\rho|m, \mathbf{f}, \Lambda)$ of the path variable given an object mapping m, floorplan $\mathbf{f}$, and lexicon parameters Λ is therefore given by the product of preposition and noun scores

$$\psi(\rho|m, \mathbf{f}, \Lambda) = \prod_{c=1}^{C} z_{d_c}(\theta_c, \gamma_c) \prod_{n=1}^{N} w_{e_n, l_{m_n}} \tag{4}$$

where c indexes into the C prepositions in the graphical model, d_c is the index in the lexicon of the cth 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 nth noun in the graphical model, and l_{m_n} is the class label of the object mapped to the nth noun.

Fig. 3 visualizes the score of the second (right) graphical model in Fig. 2. This score is plotted as a function of the position of the path variable in a 2D floorplan. Two plots are shown, with different positions for the previous path variable. The two plots have very different shapes because of differing previous positions, but both show that the highest scoring positions (in red) satisfy the intuitive meaning of the clause *toward the cone and right of the chair right of the table*. Moving from each of the two previous positions to a high scoring region results in motion *toward* the correct object and results in a position *right* of the correct object. These scores were produced by summing over all possible mappings m, using models learned automatically as described in the following.

IV. Tasks

The acquisition, generation, and comprehension tasks are formulated around the same scoring function.

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 "sentence" may be either a single sentence or possibly a sequence of sentences. The sentences and sequences of sentences are treated identically by identifying the sequence of temporal segments in the text and creating an HMM representing the sequence. Each such HMM has a state corresponding to every temporal segment t in its corresponding training sentence(s). The observations for each such HMM consist of the sequence of waypoints in the path–sentence pair. The output model R_t for each state is the graphical model constructed from that temporal segment t, given the current estimate of the parameters in Λ and marginalized over all mappings m between floorplan variables in the graphical model and objects in the floorplan

$$R_t(\rho_t, \mathbf{f}, \Lambda) = \sum_m \psi_t(\rho_t|m, \mathbf{f}, \Lambda).$$

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 first state and end in the last. Dummy states, with a fixed uniform 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. These are added because a sentence can be true without describing every portion of the path, to allow the model to score highly in such situations.

Fig. 5 illustrates the automatic construction of such an HMM from a sentence. The sentence is broken into segments, and a graphical model is created representing each segment, as described previously. 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. Dummy states are added. 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.

The HMMs are used to infer the alignment between the densely sampled points in each path and the sequence of temporal segments in its corresponding sentence. This process is illustrated in Fig. 6. An HMM is produced to represent the semantics of the sentence (the same sentence and HMM as in Fig. 5). Each HMM output model computes the score of the position and velocity at each waypoint [Fig. 6 (top)]. This reflects an estimate of how true each part of the sentence is of each portion of the path. Each state's output score is the likelihood of the associated graphical model, marginalized over all possible mappings of floorplan variables to floorplan objects. These scores, along with the HMM transition model, can be used with the forward-backward algorithm to compute the probability of the HMM being in each state [Fig. 6 (bottom)] at each point in the path. This also yields the HMM likelihood, which is an estimate of how true the entire sentence (or sequence of sentences) is of the path.

Prior to learning the word meanings, all preposition and noun distributions are random. During acquisition of such meanings, the model is iteratively updated to increase the overall HMM likelihood taken as a product over all training samples. At each iteration, this gradually concentrates the probability mass of each HMM state's preposition distributions at those angles seen at portions of the path during which that state is of high probability. It also concentrates the probability mass of the object label distributions in those bins associated with the mappings corresponding to high HMM likelihoods.

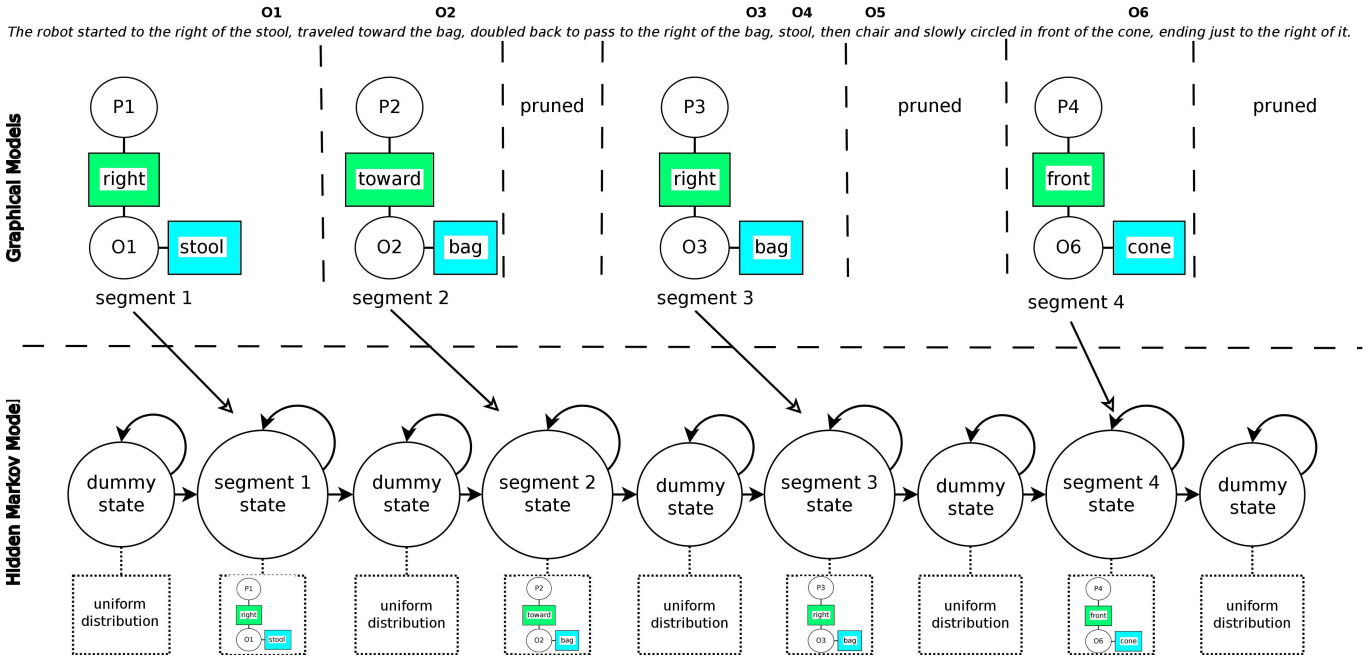


Fig. 5. HMM is created representing the semantics of a sentence. Each temporal segment of a sentence that can be parsed results in a graphical model. An HMM state is created for each such segment, with the graphical model as its output distribution. Dummy states with uniform output distributions are added before and after each state to allow it to match paths that have segments of undescribed behavior.

The robot started to the right of the stool, traveled towards 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.

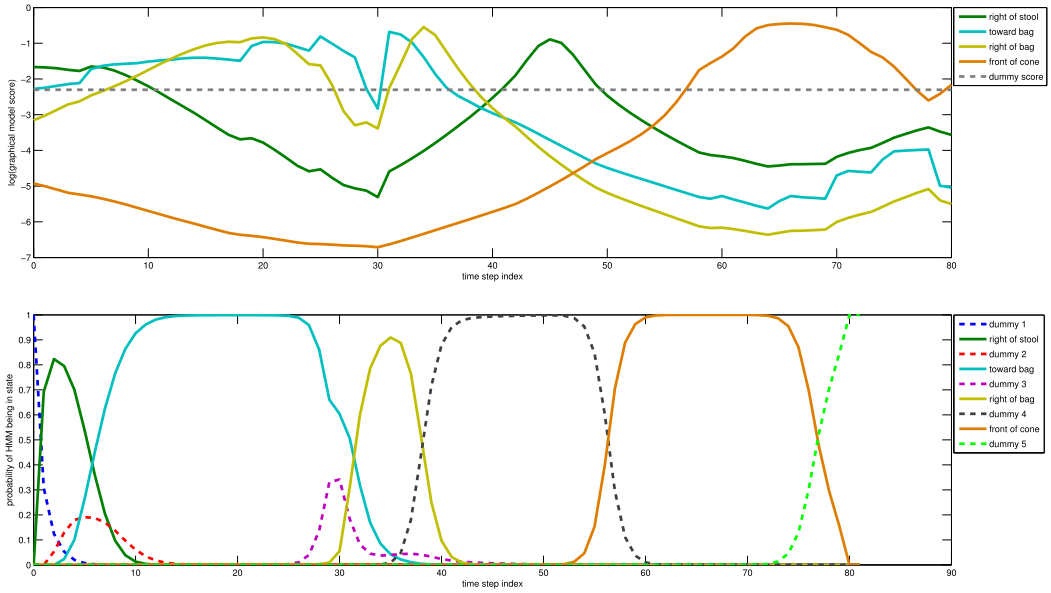


Fig. 6. Illustration of temporally aligning a sentence with a path. Top: sentence from the training set. Middle: output score of each state in the sentence-HMM computed at each point in a path. Bottom: probability of the HMM being in each state at each point in the same path.

The output models for the HMMs are all parameterized by the word meanings from the lexicon Λ. Thus, the meaning of each word is constrained by many path-sentence pairs. As illustrated in Fig. 7, 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 those relationships.

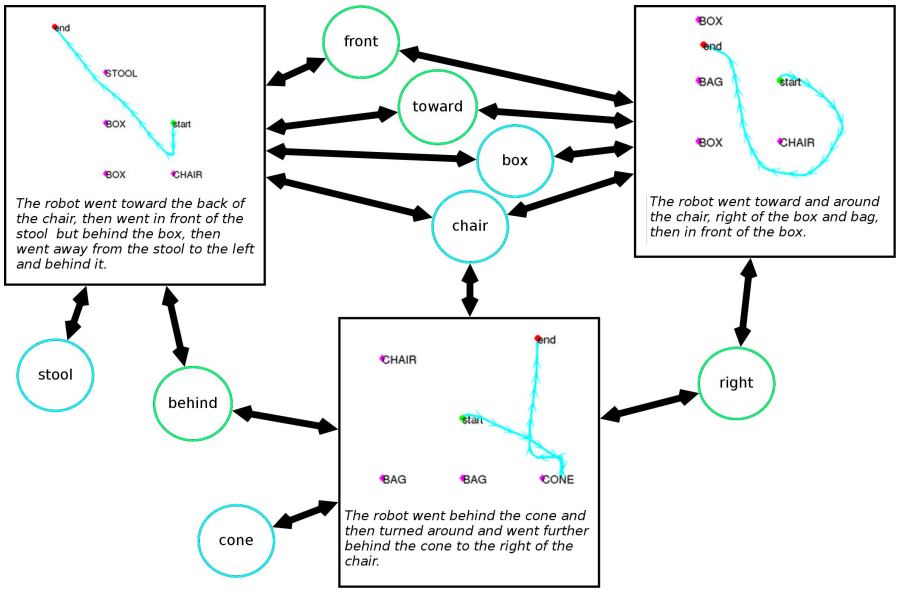


Fig. 7. Viewing the learning process as a constraint-satisfaction problem. Individual words appear across multiple path–sentence pairs. This allows inference across different words in the same sentence, where knowledge about one word constrains which points in the path or objects are described by or referred to by another. It also allows inference across multiple instances of the same word in the descriptions of different paths, where relationships among waypoints and objects in one path–sentence pair, whose description includes a particular word, constrain which relationships are referred to by that same word in the description of another path.

This learning is accomplished by maximizing the product of the likelihoods of all HMMs on their corresponding paths through Baum–Welch [2]–[4]. This trains the distributions for the words in the lexicon Λ as they are tied as components of the output models. Specifically, it infers the latent alignment between the large number of noisy robot 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. 8 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. The angular distributions of the prepositions are rendered as potential fields with points at angles with higher probability rendered lighter. Iteration 0 [Fig. 8 (top left)] shows the randomly initialized word models, and the resulting score surface, which does not encode the meaning of the phrase at all. The noun distributions are largely uniform, resulting in no visible correlation between the score and the individual object positions. After the first iteration [Fig. 8 (top right)], the noun models have just begun to concentrate in the correct bins, and the position distribution of the *right* model is beginning to concentrate in the correct direction. This change is evident in the score surface, which shows that it depends upon the position of the cone, but not in the correct way, as the *toward* model is still completely wrong. After the second iteration [Fig. 8 (bottom left)], the noun distributions are further concentrated in the correct bins, and the *toward* velocity distribution is now pointed in the correct direction, although still almost uniform. The score surface now clearly depends on both the cone and the proper chair. After the third iteration [Fig. 8 (bottom right)], the noun distributions are further concentrated, as are both the position angle distribution of the *right* model and the velocity angle distribution of the *toward* model. The score surface now largely represents the meaning of the phrase: moving from the previous position to a high scoring red region results in motion that satisfies the phrase. Iteration 3 is beginning to look similar to that in Fig. 3, which is the result after convergence.

B. Generation

To perform generation, we search for a sentence to describe a path in a floorplan. This sentence is constructed as a sequence of prepositional phrases, where the objects of the prepositions are noun phrases. The sentence is expected to satisfy three properties: 1) *correctness* that the sentence be logically true of the path; 2) *completeness* that the sentence differentiate the intended path from all other possible paths on the same floorplan; and 3) *conciseness* that the sentence be the shortest one that satisfies the previous two properties. We attempt to find a balance between these properties with the following heuristic algorithm (Fig. 9).

We first produce the most likely preposition–object pair for each waypoint. A preposition takes a waypoint from the robot path as its first argument and a floorplan object (e.g., chair) as its second argument. Thus, each preposition scores how likely the robot has a certain spatial relationship with a reference object at the current waypoint. For each waypoint, we compute the probabilities of all the possible prepositions each with all the possible reference objects on the floorplan, and select the preposition–object pair with the maximum posterior probability. This yields a sequence of selected preposition–object pairs, whose length is equal to the number of waypoints. Identical preposition–object pairs for

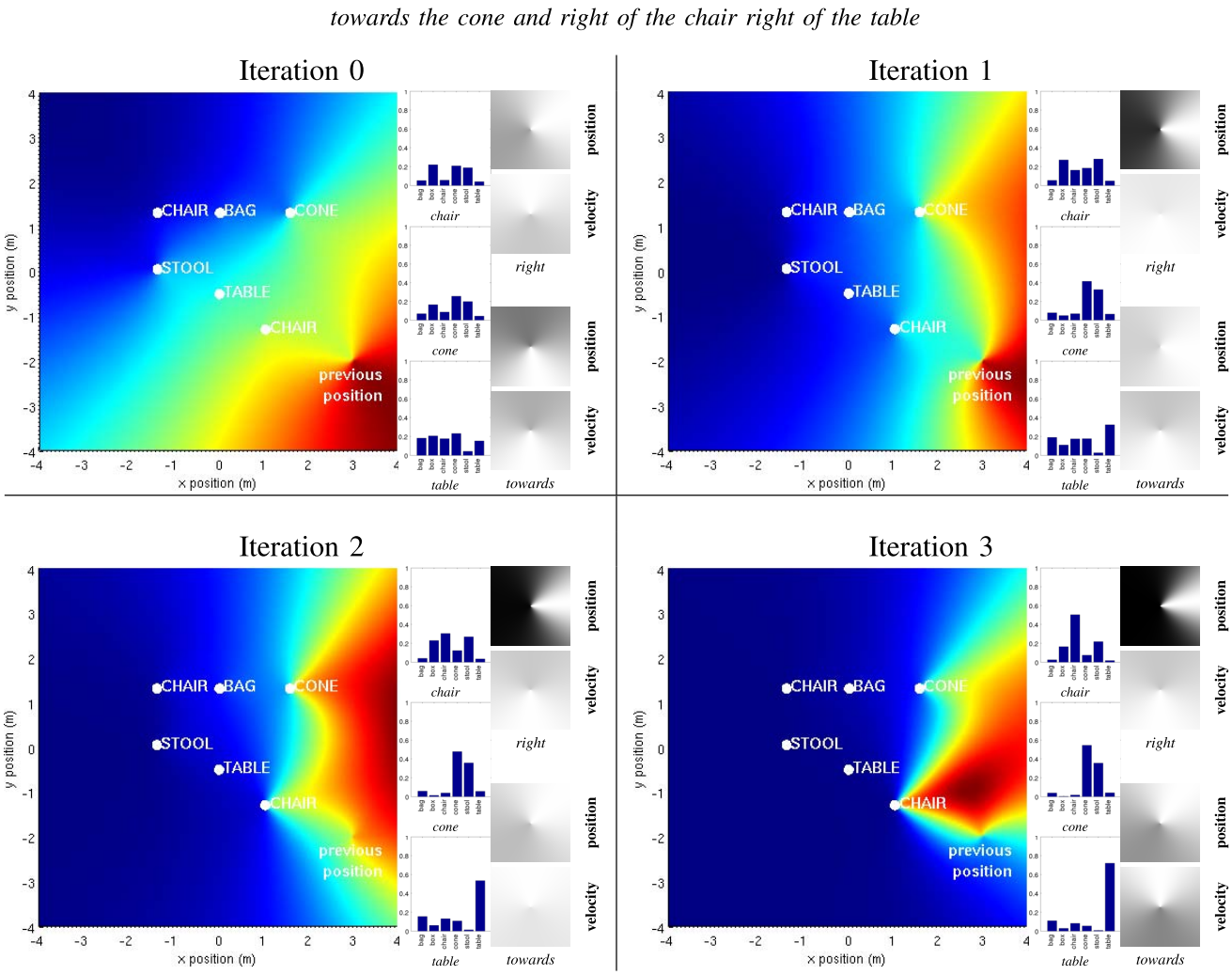


Fig. 8. Illustration of the word meanings and resulting scoring functions at the first four steps in the learning process for the same phrase illustrated in Fig. 3. The red regions of high score are initially meaningless (Iteration 0, top left), but gradually begin to capture the meaning of the phrase as the words are learned, so that moving to them from the previous robot position increasingly satisfies the phrase.

consecutive sets of waypoints in the path are coalesced into intervals, and short intervals are discarded.

We then generate a noun phrase to describe the reference object of the selected preposition–object pair at each waypoint. We take the noun with maximum posterior probability over all the possible nouns, given the class of that floorplan object. Thus, when the floorplan contains a single instance of an object class, it can be referred to with a simple noun. However, sometimes there might be more than one floorplan object that is described with the same noun. A simple noun in this case would introduce ambiguity into the generated sentence. To avoid such, the shortest possible noun phrase, with one or more prepositional phrases, is generated to disambiguate references to these objects. To this end, for each pair of floorplan objects, we take the preposition with maximum posterior probability to be true of that pair and all other prepositions applied to that pair to be false. By doing so, we assign each floorplan object with a *unique* noun phrase that is able to distinguish it from all the others on the same floorplan.

More formally, let $q(o)$ be the most probable noun for floorplan object o given Λ. For each pair (o, o') of floorplan objects, there exists only one preposition ϕ 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: 1) $u(o) = (q(o), \{\})$ is unique or 2) there exists a collection $\{\phi(o, o'), \ldots\}$ of prepositional phrases such that formula $u(o) = (q(o), \{(\phi, u(o')), \ldots\})$ is unique. To produce a concise sentence, we want the size of the collection of prepositional phrases in step 2 to be as small as possible. However, finding the smallest collection of modifiers is NP-hard [5]. To avoid exhaustive search, we use a greedy heuristic that biases toward adding the least frequent pairs $(\phi, u(o'))$ into the collection until $u(o)$ is unique. This results in a tractable polynomial algorithm. The $u(o)$ so found is mapped to a noun phrase by simple realization, for example

$$(\text{TABLE}, \{(\text{LEFTOF}, \text{CHAIR}), (\text{BEHIND}, \text{TABLE})\})$$
$$\downarrow$$
the table which is left of the chair and behind the table.

The prepositions selected for the waypoints, together with the unique noun phrases describing the corresponding reference

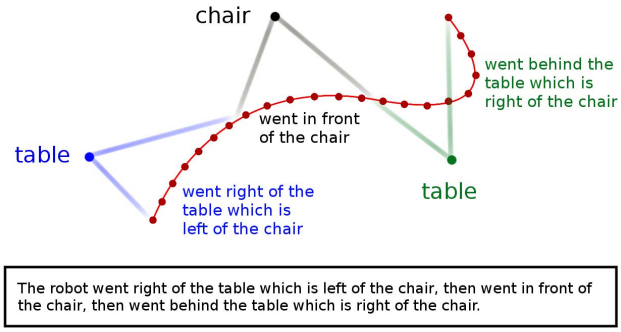


Fig. 9. Illustration of the generation algorithm. A disambiguating noun phrase is generated for each floorplan object. Waypoints are described by prepositional phrases, and then sets of identical phrases are merged into intervals, which are combined to form the sentence.

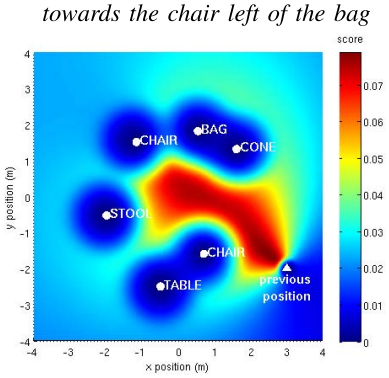


Fig. 10. Illustration of the scoring function after the addition of the barrier penalties, which keep the comprehension waypoints away from the objects and from each other, and attraction terms, which encode preference for proximity to the target object. Moving from the previous robot position to a high scoring (red) region satisfies the meaning of the phrase.

objects of the selected prepositions, are then assembled into a sentence (Fig. 9).

Generation is applied to paths obtained by odometry during human teleoperation of the robot. Such paths are sampled at 50 Hz. Because of the high sampling frequency, these paths have many redundant waypoints that provide little information to the generation process. Thus, as a preprocessing step, we downsample the path by computing the integral distance from the beginning of the path to each waypoint on the path and selecting waypoints every 5 cm of the integral length.

C. Comprehension

To perform comprehension, we use gradient ascent to find $\hat{\mathbf{p}}$ in (3), where $\mathcal{R}(\mathbf{s}, \mathbf{p}, \mathbf{f}, \Lambda)$ is the product of the graphical model likelihoods $\psi_t(\rho_t|m, \mathbf{f}, \Lambda)$ from (4) constructed from the temporal segments of the sentence $\mathbf{s}$. The unknown path $\hat{\mathbf{p}}$ is constructed to contain one path variable ρ_t for each temporal segment t in the sentence, whose locations are optimized to maximize the scoring function, and thus find waypoints that maximize the degree to which the semantics of the sentence are satisfied. This differs from pathfinding algorithms in general, where an initial point and goal point are given, and the algorithm must find dense intermediate points which avoid

The robot went behind the table in front of the bag. Next, it drove towards the table to the right of the bag, then in front of the chair that is behind the cone. Finally, it stopped to the right of the box and behind a chair.

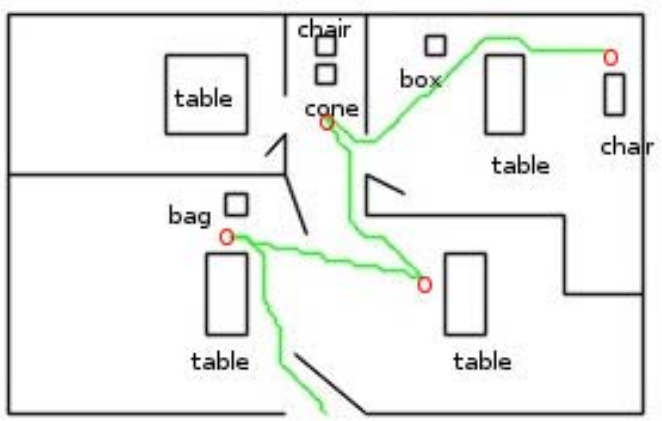


Fig. 11. Example of using a more sophisticated pathfinding algorithm, D^*, when the floorplan is complicated enough to require it. The sequence of goal points that satisfies the sentence is shown in red. These points are input to the pathfinder to avoid walls and other obstacles.

obstacles. Rather, the comprehension procedure determines the sparse (sequence of) goal point(s) that satisfy a sentence. Thus, the output of the sentence comprehension algorithm can be used as the input to any pathfinding algorithm when obstacle avoidance is needed. Fig. 11 shows an example of using the D^* [6] algorithm to perform pathfinding in a more complex environment with walls, doorways, and differently shaped obstacles.

The optimization in (3) is used to find a sparse set of waypoints that are eventually input to pathfinding. It computes a MAP estimate of the product of the likelihoods of the graphical models associated with the sentence $\mathbf{s}$. As stated, 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 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 waypoint and floorplan object as well as between pairs of temporally adjacent 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 waypoint and an object or between two 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 waypoints and objects, meant to prevent collision, both

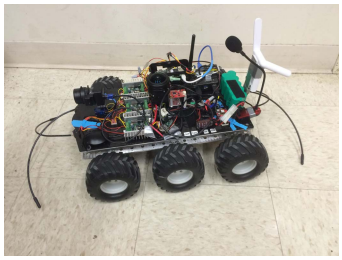


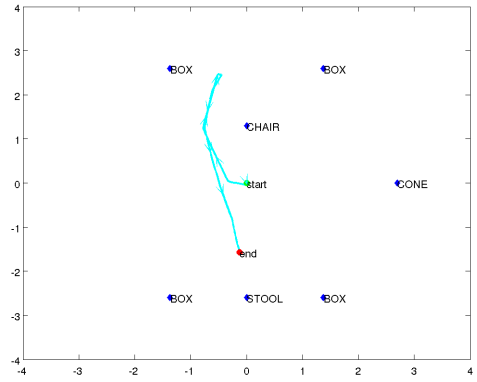
Fig. 12. Our custom mobile robot.

the robot radius and object radii are assumed to be 40 cm. For the penalty between temporally adjacent waypoints, meant to ease the optimization problem, r_1 and r_2 are set to 10 cm. 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. 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 waypoint and the floorplan objects selected as its reference objects, where r is the distance between the waypoint and the target object of a preposition. The score optimized is the product of the graphical-model factors for each waypoint along with the barrier and attraction terms. An example of the scoring function corresponding to the example phrase *toward the chair which is left of the bag*, together with the additional terms, is shown in Fig. 10. Its gradient with respect to the waypoint locations is computed with automatic differentiation [7]. The sequence of 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. They must be optimized jointly. Rather than initializing this joint optimization randomly, which we found in practice frequently resulted in the optimizer getting stuck in poor solutions, we use a multistep initialization procedure conceptually similar to layer-by-layer training in neural networks such as in [8]. The score is optimized repeatedly with subsets of the waypoints increasing in size. Beginning with the temporally first and ending with the last, waypoints are added sequentially, each time intitializing the newly added point 10 cm from the last point in the previously optimized subset. 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.

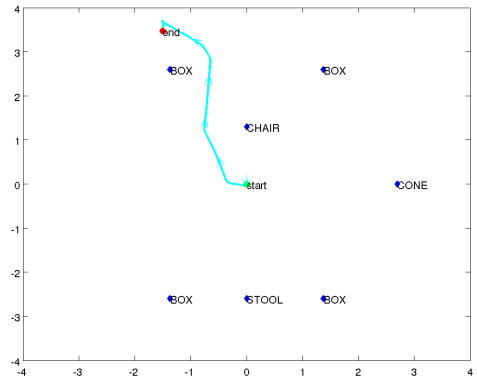
Fig. 13 shows the effect of small differences in the input sentence on the resulting paths for a series of example sentences. A single-word change can greatly alter the path by changing where the robot goes with respect to one object or by completely changing which object is referenced. This leads to changes in other portions of the path.

The output of the comprehension algorithm is a sparse set of waypoints corresponding to the temporal segments of the input sentence(s). To use these waypoints to actually drive a robot, it is necessary to perform pathfinding between them as a postprocessing step because while the barrier penalties do prevent the waypoints from being chosen close to objects, they do not prevent the paths between them from doing so.

*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.*

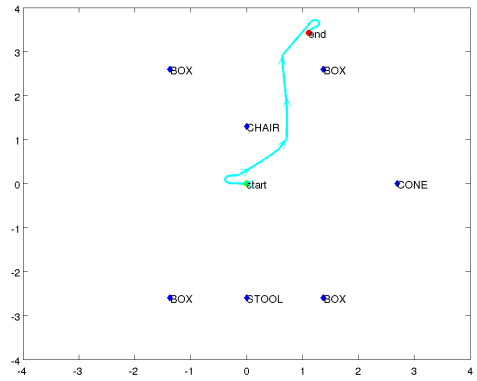


Fig. 13. Illustration of the effect on the comprehension system of single-word changes to the input sentence. The word *right* is changed to *behind* between the top and middle examples, altering where the path goes with respect to a particular box. Between the middle and bottom examples, the word *left* is changed to *right*, completely changing which box is being referred to, and therefore drastically altering the path.

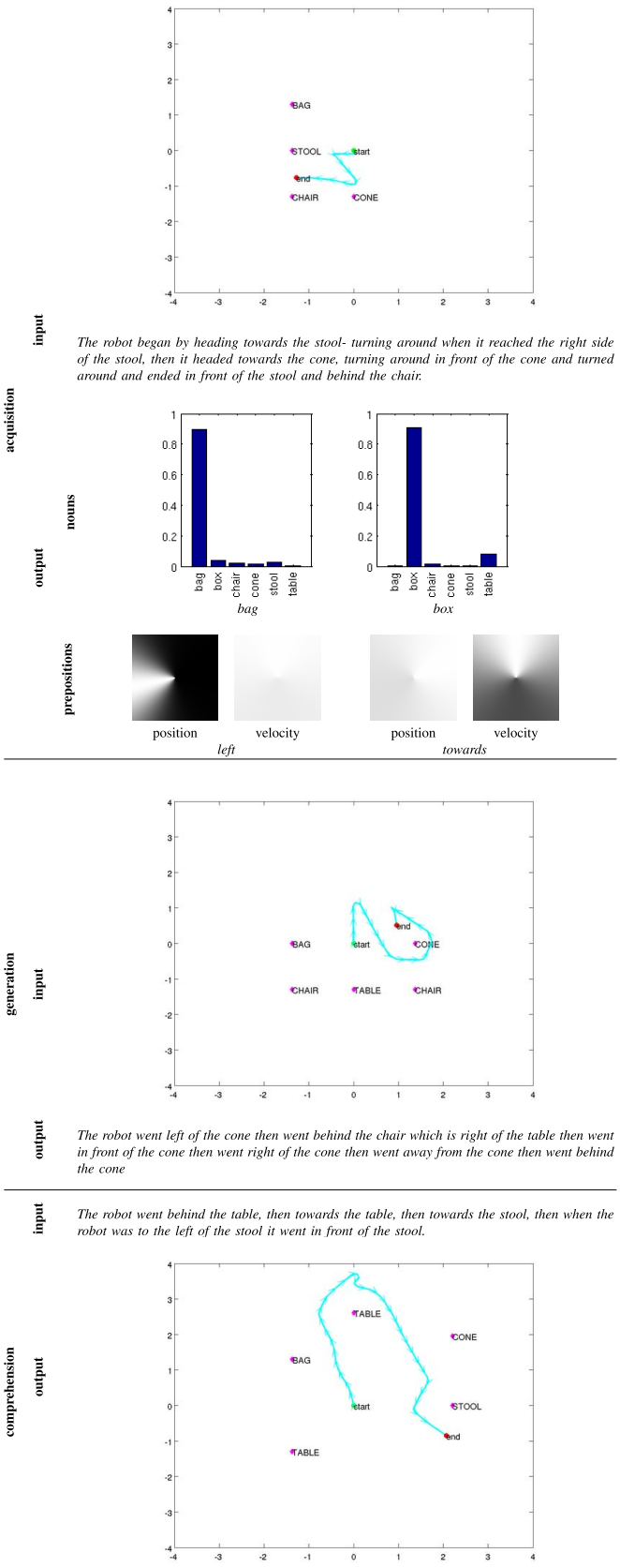


Fig. 14. Examples of the inputs and outputs for each of acquisition, generation, and comprehension.

Any path-finding algorithm with sufficient power to handle the floorplan can do. In our experiments, we used a simple procedure that recursively adds an additional waypoint to each path segment (the line segment between two goal waypoints) that passes through an obstacle. The new point is offset so that the two new path segments do not pass through the obstacle. This process is repeated recursively on new path segments until no segment passes through an obstacle.

V. Experiments

We conducted experiments to evaluate each mode of operation: acquisition, generation, and comprehension. For acquisition, we collected a corpus of robot paths paired with sentences describing those paths and used such to learn a lexicon. We then used this lexicon to test generation using a new set of paths to automatically generate sentences. In addition, we tested comprehension using the learned lexicon to produce and follow paths satisfying a new set of sentences. Fig. 14 illustrates example inputs to, and outputs from, the acquisition, generation, and comprehension systems.

All experiments were performed on a custom mobile robot (Fig. 12). This robot could be driven by a human teleoperator or drive itself automatically to accomplish specified navigational goals. During all operations, robot localization was performed onboard the robot in real time via an extended Kalman filter [9] with odometry from shaft encoders on the wheels and inertial guidance from an inertial measurement unit. This localization supported real-time path following during comprehension and was stored to support acquisition and generation. All robot paths used in our acquisition and generation experiments were obtained by recording human driving using odometry and sensor data from the robot.

A. Data Set Collection

We collected three sets of robot paths, some paired with sentences. The first, the *acquisition corpus*, consisted of 750 path–sentence pairs, three sentences for each of 250 robot paths. The second, the *generation corpus*, consisted of 100 robot paths. The third, the *comprehension corpus*, consisted of 300 path–sentence pairs, three sentences for each of 100 robot paths.

All inputs used to evaluate our method were generated by humans: the robot paths input to acquisition and generation were driven by humans and the sentences input to acquisition and comprehension were elicited from AMT workers to describe such human-driven robot paths. Each human-driven robot path was obtained by a human driving the robot in a path through a randomly generated floorplan in accordance with a randomly generated sentence. Each human-elicited sentence was obtained from an anonymous AMT worker describing such a human-driven robot path. The random synthetic sentences were only used to seed the process of generating human-driven robot paths and, in turn, human-elicited sentences. The algorithms were all applied to, or evaluated against, human sentences elicited from AMT, never synthetic sentences.

During the process of generating human-driven robot paths from random sentences, 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,

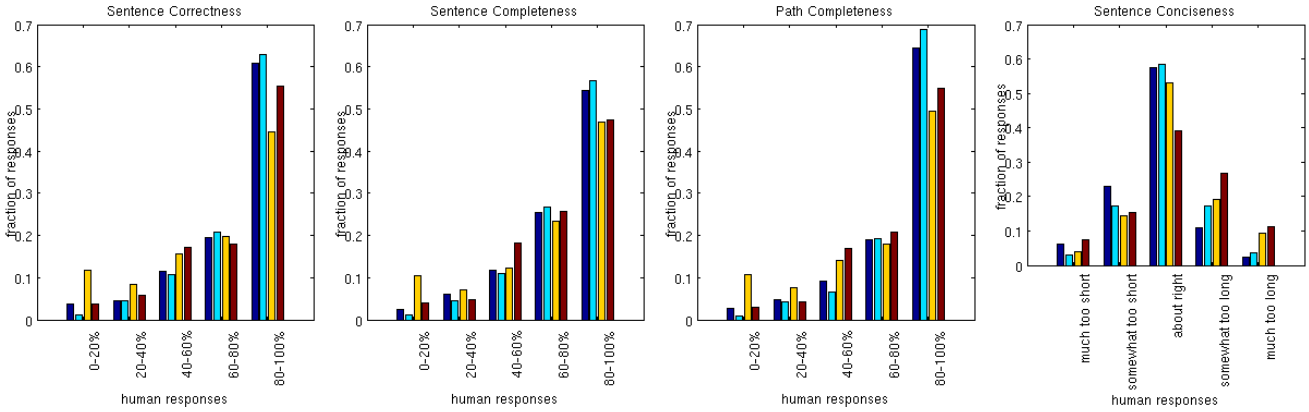


Fig. 15. Bar graphs showing the distribution of responses given by AMT workers for each of the four questions: sentence correctness (far left), sentence completeness (middle left), path completeness (middle right), and sentence conciseness (far right). The distributions are shown for sentences elicited from AMT workers and judged against the acquisition (dark blue) and comprehension (light blue) paths used to elicit the sentences, as well as for paths produced by the comprehension system judged against the human sentences used as input (yellow) and for machine-generated sentences judged against the paths used as input (red).

portions which depict the described physical relationships, the driven paths are generally more complex than the original random sentence.

The 250 human-driven robot paths for the acquisition corpus were obtained by following 25 random sentences in each of ten random floorplans. Each floorplan contained four random objects, with up to one duplicate object. The objects were placed randomly at known locations in a large room. Each random sentence contained a sequence of two or three instructions to move according to random prepositions chosen from *left*, *right*, *front*, *behind*, *toward*, and *away* with respect to randomly chosen objects in the floorplan.

The 100 human-driven robot paths for each of the generation and comprehension corpora were obtained by following ten random sentences in each of ten random floorplans. Each floorplan contained five random objects, also limited to one duplicate, which were placed randomly at known locations. The random sentences for these corpora were generated similar to those of the acquisition corpus, only longer, with a sequence of five or six instructions.

The process of eliciting human sentences from AMT was designed to yield natural and unconstrained sentences. Fig. 16 gives a small sample of these sentences. The AMT workers were asked to provide a sentence describing a robot path, as depicted in an image, relative to objects in the floorplan. They were not given any restrictions on the syntax of the sentence, nor were they told the intended purpose of the sentence. Therefore, the sentences are not artificially mechanical or specific, as they might have been had the workers known what the sentences were for. Rather, they are representative of human path descriptions in this domain. Some workers used multiple sentences to describe the path. Other workers used a single long sentence. These were used without modification.[2]

Many of the sentences elicited from AMT workers contain ambiguity, misspellings, and 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 was, 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 human-elicited sentences can be seen in Fig. 15, which summarizes the judgments by a second round of AMT workers comparing the sentences elicited from the first round of AMT workers against the paths used to elicit them (dark blue for acquisition sentences and light blue for comprehension sentences). Only about 60% of human-elicited sentences received the highest possible rating, with the rest being judged less than 80% correct or complete. Our methods are robust enough to handle such errors gracefully. The acquisition 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.

- *The robot went toward the stool which is behind the left table then to the right of the table and then went away from the stool in the direction behind the stool*
- *The robot moved toward the stool then turned toward the location to the right of the table in front of the stool then stopped at the right of the table in front of the stool.*
- *robot moved toward the stool then turned toward the location to the right of the table in front of the stool then stopped at the right of the table in front of the stool.*
- *The robot was to the right of the stool, then went behind the bag, then went to the left of the bag, then went to the left of the stool, then went to the left of the chair, then went in front of the chair.*
- *The robot went towards the cone, which is above the bag. It then turned completely around and went towards the box, passing by the starting point. He ended just to the right of the box.*
- *The robot began by going towards the table to the right. Once in front of the table to the right, it turned away from the table and headed towards the stool. The robot stopped in front of the table.*

Fig. 16. Sample human-elicited sentences obtained from AMT used in our experiments.

B. Experimental Evaluation

The acquisition, generation, and comprehension corpora were used to evaluate the acquisition, generation, and comprehension modes of operation, respectively. After learning

[2]The sentences used for expository purposes in the text, including those in Figs. 2, 3, 8–11, and 13, were constructed by hand, not elicited from AMT, and not used in our experiments. The sentences in Figs. 5–7, 14, and 16 were elicited from AMT and used in our experiments.

the meanings of the words from the acquisition corpus, the human-driven paths from the generation corpus were used to automatically produce sentential descriptions, and the human-elicited sentences from the comprehension corpus were used to automatically drive the robot. Human judgments were obtained using AMT, evaluating the degree to which the sentence and path in each pair match. Such judgments were obtained for the automatically produced sentences describing human-driven paths (e.g., generation), for the robot paths automatically driven according to human sentences (e.g., comprehension), and also for the sentences produced by AMT workers to describe human-driven paths (e.g., acquisition). For comprehension, such evaluation was performed on the automatically driven path obtained by odometry, not on the machine-generated path planned to guide such automated driving. This allowed us to compare the performance of the automatic systems with that of AMT workers through four multiple-choice questions.

1) *Sentence Correctness:* Approximately how much of the sentence is true of the path?
2) *Sentence Completeness:* Approximately how much of the path is described by the sentence?
3) *Path Completeness:* Approximately how much of the sentence is depicted by the path?
4) *Sentence Conciseness:* Rate the length of the sentence.

For the first three questions, the possible answers were 0%–20%, 20%–40%, 40%–60%, 60%–80%, and 80%–100%. This allowed us to evaluate the correctness and completeness of machine-generated paths and sentences and compare such with human performance. For the last question, the possible answers were *much too short*, *somewhat too short*, *about right*, *somewhat too long*, and *much too long*. This allowed us to evaluate the verbosity of the sentence generation system against human-elicited sentences.

We obtained three independent judgments for each path–sentence pair in order to evaluate the reliability of the human judgments. For 26.5% of the pairs, all three judgments agreed, for 54.1% of the pairs, two of the judgments agreed, and for 19.4% of the pairs, all three judgments differed.

Fig. 15 shows the distribution of judgments given to each set of path–sentence pairs for each question. The fraction of human judgments in each of the possible responses for each of the questions is shown for the output of human annotators (dark and light blue), the comprehension system (yellow), and the generation system (red), each judged against the paths or sentences given as input. For sentence conciseness, sentence length was judged about right 58.1% of the time for human-elicited sentences and 39.1% of the time for our generation system. The about right and somewhat too short/long judgments combine to 92.4% for humans and 81.3% for our generation system. Averaging the judgments for the first three questions (sentence correctness, sentence completeness, and path completeness) yields 82.4% and 85.3% for the human sentences on the acquisition and comprehension corpora, respectively, when judged against the paths used to elicit those sentences. Averaging the same judgments for the automatically driven paths judged against the human sentences used as input yields 71.1%. Averaging the judgments for the automatically produced sentences judged against the paths used as input yields 78.6%. Overall, machine performance is 74.9%, while the performance of human annotators is 83.8%. Machine performance is thus 89.2% of the way toward that of the humans.

VI. Related Work

We know of no other work which presents a physical robot that 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.

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; they utilize the internal representation of the simulation to obtain discrete symbolic primitives [10]–[17]. They have a small space of possible robot actions, positions, and states, which are represented in terms of symbolic primitives. 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 learning 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 densely sampled points in the real 2D Cartesian plane. Not all points in a path correspond to words in its sentence; multiple (often undescribed) relationships can be true of any point, and the correspondence between described relationships and path points is unknown. Furthermore, our system does not require additional manually annotated data upon which much of the previous work depends.

There has been work on learning in the context of language and mobile robot navigation using a physical robot (see [18], [19]), but none of these do all three of the tasks (acquisition, generation, and comprehension) which we do. There is also recent work on the topic of natural-language interaction with robots (e.g., [20]–[25]), both within and outside the realm of robotic navigation. However, such work does not involve any learning. There is work which learns in the context of language and robotics, but not navigation (see [26]–[29]).

VII. Conclusion

We demonstrate a novel approach for grounding the semantics of natural language in the domain of robot navigation. This approach allows the meanings of nouns and prepositions to be learned successfully from a data set of sentences describing paths driven by a robot despite substantial amounts of error in those descriptions. These learned word meanings 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. The quality of these paths and sentences averages 89.2% of the way toward the performance of human annotators on AMT. This is a step

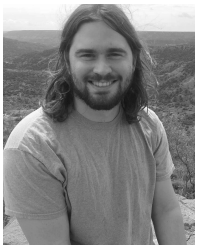
toward the ultimate goal of grounded natural language that allows machines to interact with humans when the language refers to actual things and activities in the real world.

ACKNOWLEDGMENT

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 ARL, NSF, 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

[1] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*. New York, NY, USA: Dover, 1972.
[2] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *Ann. Math. Statist.*, vol. 37, no. 6, pp. 1554–1563, 1966.
[3] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Ann. Math. Statist.*, vol. 41, no. 1, pp. 164–171, Feb. 1970.
[4] O. Shisha, Ed. "Inequalities III," *Proc. 3rd Symp. Inequal*. New York, NY, USA, pp. 1–9. Sep. 1972.
[5] R. Dale and E. Reiter, "Computational interpretations of the Gricean maxims in the generation of referring expressions," *Cognit. Sci.*, vol. 19, no. 2, pp. 233–263, Apr. 1995.
[6] A. Stentz, "The focussed D* algorithm for real-time replanning," in *Proc. Int. Joint Conf. Artif. Intell.*, vol. 95. Aug. 1995, pp. 1652–1659.
[7] A. Griewank, "On automatic differentiation," in *Mathematical Programming: Recent Developments and Applications*, M. Iri and K. Tanabe, Eds. New York, NY, USA: Academic, 1989, pp. 83–108.
[8] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.
[9] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. New York, NY, USA: Academic, 1970.
[10] M. MacMahon, B. Stankiewicz, and B. Kuipers, "Walk the talk: Connecting language, knowledge, and action in route instructions," in *Proc. AAAI Conf. Artif. Intell.*, 2006, pp. 1475–1482.
[11] T. Kollar, S. Tellex, D. Roy, and N. Roy, "Toward understanding natural language directions," in *Proc. Int. Conf. Human-Robot Int. (HRI)*, Mar. 2010, pp. 259–266.
[12] C. Matuszek, D. Fox, and K. Koscher, "Following directions using statistical machine translation," in *Proc. ACM/IEEE Int. Conf. Human-Robot Int. (HRI)*, Mar. 2010, pp. 251–258.
[13] S. Tellex *et al.*, "Understanding natural language commands for robotic navigation and mobile manipulation," in *Proc. AAAI Conf. Artif. Intell.*, Aug. 2011, pp. 1507–1514.
[14] D. L. Chen and R. J. Mooney, "Learning to interpret natural language navigation instructions from observations," in *Proc. AAAI Conf. Artif. Intell.*, Aug. 2011, pp. 859–865.
[15] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox, "Learning to parse natural language commands to a robot control system," in *Proc. Int. Symp. Experim. Robot.*, 2013, pp. 403–415.
[16] Y. Artzi and L. Zettlemoyer, "Weakly supervised learning of semantic parsers for mapping instructions to actions," *Trans. Assoc. Comput. Linguistics*, vol. 1, no. 1, pp. 49–62, Mar. 2013.
[17] S. Tellex, P. Thaker, J. Joseph, and N. Roy, "Learning perceptually grounded word meanings from unaligned parallel data," *Mach. Learn.*, vol. 92, no. 2, pp. 151–167, Feb. 2014.
[18] "S. Pulman and A. Harrison, "Teaching a robot spatial expressions," in *Proc. 2nd ACL-SIGSEM Workshop Linguist. Dimensions Prepositions Comput. Linguist. Formal. Appl*. Colchester, U.K., Apr. 2005, pp. 19–21.
[19] S. Lauria, G. Bugmann, T. Kyriacou, and E. Klein, "Mobile robot programming using natural language," *Robot. Auto. Syst.*, vol. 38, nos. 3–4, pp. 171–181, Mar. 2002.
[20] S. Teller *et al.*, "A voice-commandable robotic forklift working alongside humans in minimally-prepared outdoor environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2010, pp. 526–533.
[21] A. Koller *et al.*, "Report on the second NLG challenge on generating instructions in virtual environments (GIVE-2)," in *Proc. 6th Int. Natural Lang. Genera. Conf.*, Jul. 2010, pp. 243–250.
[22] T. K. Harris, S. Banerjee, and A. I. Rudnicky, "Heterogeneous multi-robot dialogues for search tasks," in *Proc. AAAI Spring Symp. Dial. Robots Verbal Interaction Embodied Agents Situated Devices*, Stanford, CA, USA, Mar. 2005, pp. 21–23.
[23] M. Marge, A. Pappu, B. Frisch, T. K. Harris and A.I. Rudnicky, "Exploring spoken dialog interaction in human-robot teams," *Robots, Games, Res. Success Stories USARSim IROS Workshop*, 2009.
[24] A. Pappu and A. Rudnicky, "The structure and generality of spoken route instructions," in *Proc. 13th Annu. Meeting Special Interest Group Discourse Dialogue*, Jul. 2012, pp. 99–107.
[25] J. Fasola and M. J. Mataric, "Using semantic fields to model dynamic spatial relations in a robot architecture for natural language instruction of service robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 143–150.
[26] P. McGuire *et al.*, "Multi-modal human-machine communication for instructing robot grasping tasks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 2. Oct. 2002, pp. 1082–1088.
[27] F. Doshi and N. Roy, "Spoken language interaction with model uncertainty: An adaptive human–robot interaction system," *Connection Sci.*, vol. 20, no. 4, pp. 299–318, Nov. 2008.
[28] C. Matuszek, N. FitzGerald, L. Zettlemoyer, L. Bo, and D. Fox, "A joint model of language and perception for grounded attribute learning," in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 1671–1678.
[29] L. She, S. Yang, Y. Cheng, Y. Jia, J. Y. Chai, and N. Xi, "Back to the blocks world: Learning new actions through situated human-robot dialogue," in *Proc. 15th Annu. Meeting Special Interest Group Discourse Dialogue (SIGDIAL)*, 2014, pp. 89–97.



Daniel Paul Barrett (M'17) received the B.S.Cmp.E. degree from Purdue University, West Lafayette, IN, USA, in 2011, and the Ph.D. degree from the School of Electrical and Computer Engineering, Purdue University, in 2016.

He is currently a Research and Development Scientist with Sandia National Laboratories, Albuquerque, NM, USA. His current 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 (M'08) received the B.S. degree in electrical and computer engineering from the U.S. Military Academy, West Point, NY, USA, in 1999, the M.S. degree in electrical and computer engineering from Kansas State University, Manhattan, KS, USA, in 2009, and the Ph.D. degree from the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA, in 2016.

From 2009 to 2012, he was an Instructor and an Assistant Professor with the Department of Electrical Engineering and Computer Science, U.S. Military Academy. He is currently a Senior Systems Engineer with General Motors, Milford, MI, USA. His current 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 (M'17) received the B.S. degree in computer science from Peking University, Beijing Shi, China, in 2011, and the Ph.D. degree from the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA, in 2016.

He is currently a Research Scientist with Baidu Research, Sunnyvale, CA, USA. His current research interests include computer vision and natural-language processing.

Dr. Yu was a recipient of the Best Paper Award of ACL 2013.

Jeffrey Mark Siskind (M'98–SM'06) received the B.A. degree in computer science from the Technion–Israel Institute of Technology, Haifa, Israel, in 1979, the S.M. and Ph.D. degrees in computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1989 and 1992, respectively.

He was a Post-Doctoral Fellow with the Institute for Research in Cognitive Science, University of Pennsylvania, Philadelphia, PA, USA, from 1992 to 1993. From 1993 to 1995, he was an Assistant Professor with the Department of Computer Science, University of Toronto, Toronto, ON, Canada. He was a Senior Lecturer with the Department of Electrical Engineering, Technion–Israel Institute of Technology, in 1996. He was a Visiting Assistant Professor with the Department of Computer Science and Electrical Engineering, University of Vermont, Burlington, VT, USA, from 1996 to 1997. He was a Research Scientist with NEC Research Institute Inc., Princeton, NJ, USA, from 1997 to 2001. He has been an Associate Professor with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA, since 2002.