Following Directions Using Statistical Machine Translation

Cynthia Matuszek, Dieter Fox, Karl Koscher
Computer Science & Engineering
University of Washington
Seattle, WA 98195-2350
{cynthia, fox, supersat}@cs.washington.edu

***Abstract*—Mobile robots that interact with humans in an intuitive way must be able to follow directions provided by humans in unconstrained natural language. In this work we investigate how statistical machine translation techniques can be used to bridge the gap between natural language route instructions and a map of an environment built by a robot. Our approach uses training data to learn to translate from natural language instructions to an automatically-labeled map. The complexity of the translation process is controlled by taking advantage of physical constraints imposed by the map. As a result, our technique can efficiently handle uncertainty in both map labeling and parsing. Our experiments demonstrate the promising capabilities achieved by our approach.**

***Index Terms*—Human-robot interaction; instruction following; navigation; statistical machine translation; natural language**

I. INTRODUCTION

Following natural-language route instructions through a building is a challenging, error-prone activity. Maps of spaces are often incomplete or inaccurate, and the variety of ways a single path can be described is enormous. Furthermore, people are surprisingly poor at giving directions. Instructions may be ambiguous or incorrect; people confuse left and right, omit instructions for important decision points, or fail to mention intervening decision points or landmarks. Nonetheless, giving and following directions naturally is a crucial aspect of smooth human/robot interactions.

The problem of following instructions can be described as finding a way of going from the natural language description, or layer, to an underlying *map layer* that is grounded in a map built by a robot. Although there are a number of ways to consider this problem, our approach is to apply statistical machine translation (SMT) to the task of translating from natural language into a formal intermediate *path description language*, designed to closely mimic what current robotic sensors and actuators can handle in a real-world environment. This has the potential to reduce intermediate translation steps and dependence on heuristic handling of input, while minimizing assumptions about available information.

There exists a substantial body of work on robotic navigation, and specifically on direction-following in navigation and other tasks, which is described in more detail in Section IV. We provide the following additional contributions: Initial demonstration of a system that takes full advantage of existing robot mapping and place-identification technology, by showing pathfinding in an environment where available information is limited to automatically obtained map segmentation and labels—neither human labeling nor manual landmark identification are required. Our approach takes uncertainty in both the map and the language parsing into account by combining the two into a single analysis of the most likely interpretation of a set of directions. Finally, we show how to incorporate map information into the parsing process, so as to manage the combinatorics of a very non-restrictive formal language with an extraordinarily high degree of ambiguity.

To achieve this, we use existing topological place-labeling techniques [7], which enable robots to autonomously build maps that can be described in terms of *typed* areas (such as hallways, rooms, and intersections). We define a formal language that describes movement through a sequence of such areas, and a statistical machine translation (SMT) system is used to learn a probabilistic translation model between natural language and the formal description of paths. The SMT system is trained on examples of natural language route directions and the corresponding paths through a map built by a mobile robot. Because natural language directions are very ambiguous with respect to the types of areas being traversed, the space of possible movements through the map is used to constrain the parsing process, which is otherwise infeasibly large. The map uncertainty and parser probabilities are used to find a path that corresponds to the spoken directions.

This paper is organized as follows. In the next section, we present our approach, starting with some background on statistical machine translation. Experimental results are given in Section III, followed by a discussion of related work in Section IV. We conclude in Section V.

II. APPROACH

Our approach involves using machine translation to learn to parse natural language instructions into a sequence of instructions that can be executed by a robot equipped only with a map and a laser range-finder. Specifically, there are three representations of navigation through a map: statements in NL (the natural language route instructions), statements in a formal path description language, and an actual path through a map at the map layer.

We use SMT to learn a parser that transforms NL instructions into the path description layer (parsing), which can then be transformed into the map layer (path-finding). Both parsing and path-finding must handle uncertain information.

This work was partially supported by ONR MURI grant #N00014-07-1-0749 and by the NSF under award number IIS-0812671.

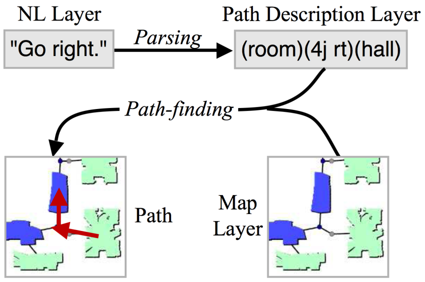


Fig. 1. The stages of converting a statement in natural language to a path through a map. Parsing is performed by a parser trained on example route instructions and map traces.

The natural language instructions are constrained by the maps provided, which do not contain landmarks, and are further limited by our grammar's inability to express *context* (information about the nodes surrounding those that are actually traversed). These constraints are not a fundamental limitation of our proposed technique, but rather a limitation of our current path description language; the impact of these simplifications is discussed in Section III and Section V.

A. LABELED MAPS AND FORMAL LANGUAGE

1) Labeled Topology Using Voronoi Random Fields: In this paper, the maps being used for training and testing are constructed from laser range-finder data, and have been automatically segmented and labeled using *Voronoi Random Fields* [7]. In this approach, a Voronoi graph is initially extracted from an occupancy grid generated with a laser range-finder; each point on the graph is represented as a node of a conditional random field, which is a discriminatively trained graphical model. The resulting VRF estimates the label of each node, integrating features from both the map and the Voronoi topology. The labels provide a segmentation of an environment, with the different segments corresponding to rooms, hallways, intersections, or doorways (as in Figure 3 (a)).

These maps can then be segmented to provide topological-metric maps of the labeled Voronoi graph. The spatial layout of rooms and hallways is retained, along with connectivity structure based on identification of intersections. This structure can be readily transformed to an idealized map of typed nodes with connecting edges; the stochastic classification is not perfect, but has an accuracy above 90% [7].

The synchronous context-free grammar (SCFG) parsing model (see Section II-B) requires that the target language (the path description layer) be defined by an unambiguous grammar (Figure 2). Furthermore, path-finding—going from the path description layer to the map layer—must be robust against uncertainty in the segmenting and labeling of the map. The path description layer devised for this process is quite simple: a path is defined as a series of nodes through the topological graph-map, connected by edges. Junction nodes are parameterized with available exits and information about the orientation of the robot as it traverses the junction, which captures rotation. Nodes are described relative to the agent's point of view, rather than in absolute terms; for example, a 3-way junction may be a T-junction, meaning there are openings to the robot's right and left, but could not be described as having openings to the east and west.

The grammar contains both *terminals*—the lexicon of words in the path description language—and *nonterminals*, which can be decomposed into terminals or other nonterminals. Nonterminals in Figure 2 are explicitly marked with `n:`, such as n:Action, whereas terminals (such as `rt` or `room`) are are not marked. While the names of the terminals are chosen to be human-readable, it should be noted that there is no semantic information encoded; the connection between the word "right" and the terminal `rt` is entirely learned.

```
n:Statement → ({ ( go n:Action ) })
n:Action → ({ n:Vertex n:Movement })
n:Movement → ({ n:Edge n:Vertex })
n:Movement → ({ n:Edge n:Vertex n:Movement })
n:Vertex → ({ ( room ) })
n:Vertex → ({ ( hall ) })
n:Edge → ({ ( n:Junction ) })
n:Junction → ({ n:Jtype n:Jdir })
n:Junction → ({ n:Jtype n:Jorient n:Jdir })
n:Jtype → ({ j4 })
n:Jtype → ({ j3 })
n:Jorient → ({ t })
n:Jorient → ({ r })
n:Jorient → ({ l })
n:Jdir → ({ rt })
n:Jdir → ({ st })
n:Jdir → ({ lt })
```

Fig. 2. The grammar of the path description language into which natural language directions are parsed. Hallways and rooms are treated as nodes of a map, connected by intersections, which are three-way or four-way junctions (`j3` and `j4`); the direction in which a junction is traversed is given as right, straight, or left (`rt`, `st`, and `lt`); and the orientation of three-way junctions is given (`t`, `r`, and `l` indicate which direction is traversible). The key "n:" is used to mark nonterminals.

2) Formal Language Path Descriptions: This statistical approach allows this system to handle noise (such as previously unseen or irrelevant words), as well as avoiding any explicit definition of structure such as labeling parts of speech or pre-defining the words that correlate to actions such as turning.

As mentioned, we make two noteworthy simplifications to the language: we do not represent *landmarks* (other than the nodes themselves) or *context*. Landmarks are any element of the environment that could be used to describe an area, such as "go past the couch" or "the room with the blue rug," while context describes nodes on the map that surround the path, but are not actually traversed. Examples of contextual path descriptions might include sentences such as "Go to the end of the hall," which requires information about the step one past the steps actually taken, or "go past a hall, then a room." In these cases, the spaces being described are not on the path taken by the agent, and so are not represented in the path descriptions, making it impossible to learn a translation into the path description language.

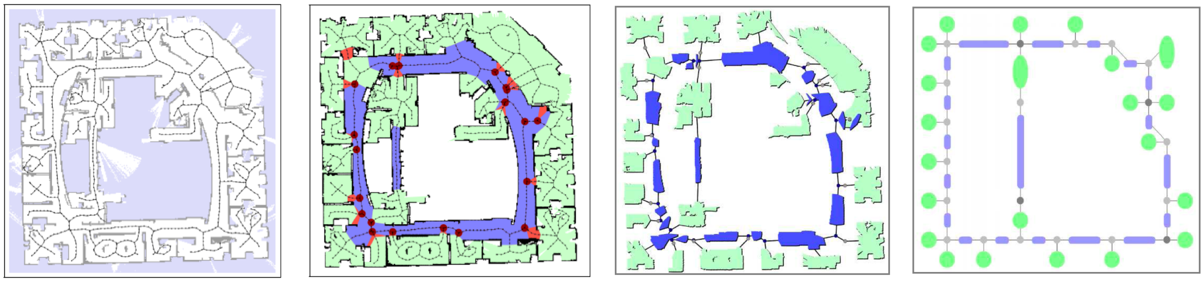


Fig. 3. The test map used for evaluation. (far left) An occupancy map constructed from laser range-finder data, with a Voronoi-based route graph shown. (left) The labeled Voronoi graph defines a place type for each point on the map. Hallways are colored red, rooms are green, and junctions are indicated by blue circles. (right) A topological-metric map given by the segmentation of the labeled Voronoi graph. (far right) The topological metric map transformed into the typed-node representation, with junctions representing edges between rooms and halls.

B. *STATISTICAL MACHINE TRANSLATION*

The underlying parsing approach used is Statistical Machine Translation, in which a large body of example sentence pairs are used to learn a translation system automatically [10]. We use a modified version of the Word Alignment-based Semantic Parser (WASP) developed by Wong & Mooney [23], which learns a parser that translates from novel strings in a source language to a formally defined target language.

WASP implements a synchronous translation learning model, meaning that parse trees are constructed simultaneously for sentences in the source and target language by applying a sequence of production rules to higher-level *nonterminals*. For example, in translating between two natural languages, the nonterminals `NP` and `V` might refer to noun phrases and verbs; a production would describe how `NP` could be synchronously replaced by the strings "I" in English and "*je*" in French.[1]

The learning process can be described at a high level as follows. First, a statistical word alignment model [3] is learned. In word alignment, pairs of corresponding strings between source and target languages are discovered, for example, *je*/I or "turn right"/`rt`. For this task the off-the-shelf word-alignment tool GIZA++ [14] is used. A lexicon of production rules is then generated by creating a rule for each aligned string pair in *each* training pair; for example, the paired training sentences "I will" and "*je vais*" provide evidence for the existence of the production rules "*je*←`NP`→I" and "*vais*←`V`→will." If sufficient training data is present, the set of production rules learned will implicitly define the set of all possible derivations of target strings. Finally, a set of parameters is learned that define a probability distribution over derivations (a sequence of rule applications). This is a probabilistic extension of the synchronous context-free grammar (SCFG) parsing framework [1].

A probabilistic SCFG $\mathcal{G}$ is then defined as follows:

$$\mathcal{G} = < \mathcal{N}, \mathcal{T}_{in}, \mathcal{T}_{out}, \mathcal{L}, \mathcal{S}, \lambda >$$

In which $\mathcal{N}$ is a finite set of nonterminals, $\mathcal{T}_{in}$ and $\mathcal{T}_{out}$ are finite sets of terminals (words) in the source and target languages, $\mathcal{L}$ is a lexicon of production rules, $\mathcal{S}$ is a defined start symbol, and λ is the set of parameters defining a probability distribution over derivations.

WASP uses a log-linear probabilistic model described by a set of parameters λ. The model applies to the probability of a particular derivation (a sequence of translation steps); the probability of a derivation $\mathbf{d}$ given an input sentence $\mathbf{e}$ is:

$$\Pr_{\lambda}(\mathbf{d}|\mathbf{e}) = \frac{1}{Z_{\lambda}(\mathbf{e})} \exp \sum_{i} \lambda_i f_i(\mathbf{d})$$

Where f_i is one of three types of features:

- For a rule, the number of times that rule is used in any derivation.
- For a word, the number of times that word is generated from a word gap in a derivation.
- The total number of words generated from word gaps.

However, only the results of derivations—that is, input/output sentence pairs $(\mathbf{f_j}, \mathbf{e_j})$—are provided as training data. We can compute the conditional probability of an output sentence being produced by an input sentence by summing over all the derivations that perform such a translation:

$$\Pr_{\lambda}(\mathbf{f_j}|\mathbf{e_j}) = \sum_{\mathbf{d}} \Pr_{\lambda}(\mathbf{d}|\mathbf{e})$$

WASP uses a variant of the Inside-Outside algorithm to avoid enumerating all possible derivations, since the number of derivations may be intractably large. An estimate of λ can then be obtained by maximizing the conditional log-likelihood of the training set, i.e. maximizing the sum of the conditional log likelihoods of each sentence pair, using gradient-descent techniques (L-BFGS in WASP's case). A detailed description of this approach can be found in [22].

The advantages of using this approach to parser learning are twofold. First, rules with nested non-terminals can readily represent commands that do not specify a fixed number of actions, such as "take the second left," which may traverse an arbitrary number of map nodes. Second, it minimizes commitment to a specific domain. This approach can be applied to any domain for which a simple grammar can be defined and examples can be readily expressed using that grammar. However, because the process of parsing from route instructions to path descriptions is not dependent on a specific map, and because the formal language being parsed into does not have complex or deeply nested structural constraints, the parse ambiguity is very large compared to parsing problems

[1] [5] is recommended to the reader seeking a clear overview of SCFGs; we follow the terminology of [22] in describing grammars, rules, and derivations.

which target more constrained formal languages such as [22], [4], [6] (see Figure 5 for an example); further work is required to manage this complexity.

C. PATH SEARCH

The parsing process produces uncertain information; additionally, because the map is automatically labeled, there is uncertainty in the annotation of the underlying map graph. (For instance, the robot might not be sure if a certain node in the topological graph is a hallway or a room.) This combined uncertainty means that evaluation of parses against the map is not necessarily reliable. Because we wish for our direction-following approach to be robust in the face of such uncertainty, we backtrack to lower-probability paths if the most probable path does not lead to the desired destination. As a result, all parses of a sequence of route instructions must be retained and considered when choosing the most probable path to take, leading to an exponential blowup in possible path traces for a given input description. The large fan-out of possible parses makes a brute-force approach to this problem intractable (see Figure 6 for example parses of a single segment).

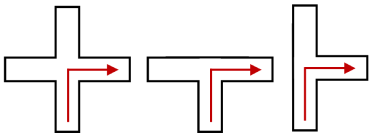


Fig. 5. An example of parsing ambiguity. Without reference to a specific map, all of the following are correct parses of the phrase "turn right," with the distinct path descriptions `(j4 rt)`, `(j3 t rt)` , and `(j3 r rt)`. Combinations of such commands (such as "turn right, then turn right again") quickly become intractable.

We treat complexity in the parsing stage and in the path-finding stage separately.

1) Parse Complexity: Route instructions may be thought of as a sequence of semi-independent segments, each of which describes a separable set of steps between decision points. Because the parse grammar does not have much nested structure, it is rare for the parse of a segment to rely on other segments; as a result, segments of the natural language inputs can generally be parsed separately. For example, given the instructions "Go left, then turn right," the meaning of "turn right" is unlikely to depend on the meaning of "Go left"—unlike unconstrained natural language, in which a later phrase may refer back to the subject of a previous phrase. Segmentation can be done either automatically (e.g., by relying on phrase boundaries learned by the parser [4]), or by pre-processing route instructions in some way; in our experiments, we segmented instructions by splitting on defined keywords.

`(room) (4j rt) (room)`	`(room) (4j rt) (hall)`
`(room) (3j r rt) (room)`	`(room) (3j r rt) (hall)`
`(room) (3j t rt) (room)`	`(room) (3j t rt) (hall)`
`(hall) (4j rt) (room)`	`(hall) (4j rt) (hall)`
`(hall) (3j r rt) (room)`	`(hall) (3j r rt) (hall)`
`(hall) (3j t rt) (room)`	`(hall) (3j t rt) (hall)`

Fig. 6. All possible ways to describe a right turn, as specified by the grammar of the path description language.

In this terminology, the natural language route instructions "Go out of the room and go right, then go past two intersections, then turn left" contains four segments: "Go out of the room; go right; go past two intersections; turn left." The correct path description when these instructions are applied to Figure 3 is `(room) (j4 rt) (hall) (j3 r st) (hall) (j3 l st) (hall) (j3 t lt) (room)`, corresponding to a path going from the entryway down to the room on the lower right. However, without oracular knowledge of the correct parse and correct map labeling, this is only one many thousands of path descriptions that must be considered.

2) Pathfinding Complexity: While segmenting controls the complexity of the parsing step, the path-finding step must still consider every possible combination of segment parses. In addition, each *step* in a parse has an associated uncertainty, produced by combining the parse weight with the current belief as to whether the step is correct with respect to the map. An example helps ground this description. A single parse of the phrase "go right" may be `(room) (j4 rt) (hall)`, with an associated parse weight of 0.5. If this command is given while the robot is at node 418 on Figure 7(a), the first step, `(room)`, is believed to be consistent with the map, and so should be scored highly. The second step, however, describes a 4-way intersection, when it is in fact believed to be a 3-way intersection; that parse receives a lower score. These scores, when combined with the parse score, must be treated individually. Sequences of nodes can be stored in a weighted tree (Figure 4 shows such a tree for the instruction "go right"); however, the size of that tree rapidly becomes intractable for larger sequences of commands.

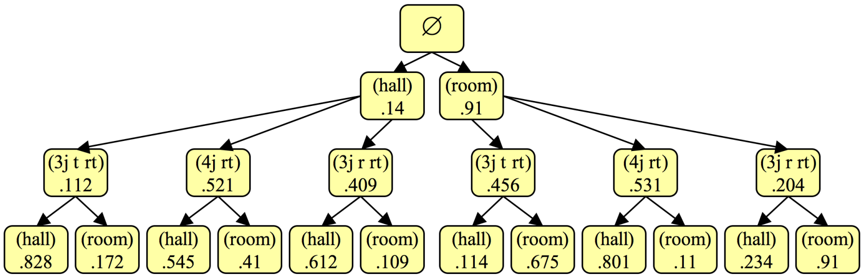


Fig. 4. The complete set of possible correct parses of the command "go right," with weights provided by the agent's beliefs about the map. Because instruction-giving is often noisy, incorrect parses should also be considered with low weight, making the fan-out of the actual tree larger.

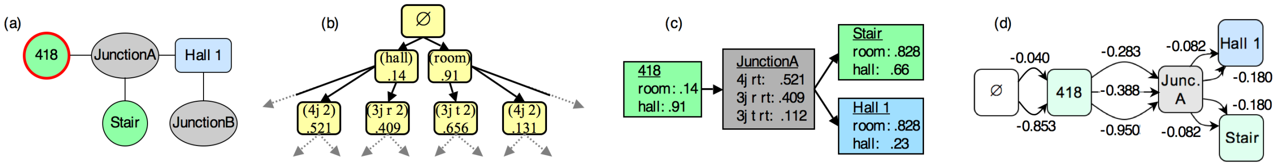


Fig. 7. The components of the tractable search problem. (a) A small section of a map through which a right turn is to be taken, starting from room 418 (outlined). (b) A portion of the parse tree showing possible interpretations of the command "turn right," with the combined probabilities of the parses and map node types. (c) The collapsed map/parse tree. (d) The search tree used to iteratively find the k-shortest paths from the source to possible destinations.

In order to search this space feasibly, we first introduce a combined map/parse tree, in which the structure of the map is used to both constrain and efficiently store possible parses. We construct a tree of nodes corresponding to regions of the map, rooted in the robot's starting position, disallowing cycles by terminating a branch when the only possible step is a node that already exists on that branch. Because there are a sharply limited number of map nodes that are n steps from a known starting point, this tree is feasible to construct. The n-th nodes of possible parses are then stored as a list attached to each tree node (see Figure 7 (c)). This representation is substantially more compact than the full tree.

We assume that the robot is in possession of a complete map, the ability to follow the path through the map, and a starting point on the map, but not a goal-point in the map. We additionally supply a test for whether the destination has been reached once a path has been traversed. Our goal is to produce an ordered list of paths to attempt to take through the map. If the first proposed path does not reach the desired destination, the next highest-probability path is selected and the robot backtracks to the point where the next proposal diverges from the path already traveled before exploring the new path.

If a path fails, it is often not obvious at what step the problem occurred, meaning it is not obvious how to prune the tree. The need to try several paths (due to map or parsing errors, or due to human errors in giving the directions) therefore means finding the first, second, ...k-th highest-probability walk through list entries on the map/parse tree. This task can be considered as a variant of the well-studied problem of finding the k-th best (lowest cost) path through a graph. We treat the individual steps as edges between nodes, with a cost proportional to their probability; Figure 7(d) shows this new graph.

The canonical solution to this task is Yen's Kth-shortest-path algorithm [24], which is $O(kn(m+n\log n))$ for a graph with n nodes and m edges; [12] provides a more efficient variant which we use for this work, with some modifications. First, because probabilities are combined via multiplication and path costs are combined by addition, we use the log of the probabilities, negated in order to make higher probabilities consistent with lower costs. Second, the Kth-shortest-path algorithm does not handle multiple edges between nodes, so dummy nodes (not shown in Figure 7) are inserted between each pair of map/parse tree nodes. Each dummy node contains a pointer to the actual step value (for example, `room`), which makes reconstructing a path description from a graph walk trivial.

Because we do not know the desired endpoint, it is necessary to find the k best paths from the start node to all other nodes in the graph, making the actual complexity $O(kn2(m+n\log n))$ in the worst case. When this graph has been constructed, the search process proceeds by creating a list of the best (k=1) path to every node, ordering the resulting $n-1$ paths by total probability, and exploring the first. If the first fails, k is incremented, $n-1$ more paths are generated (the second-best path to every node except the starting point), and the process is repeated.

III. Evaluation

As described in Section II-C, we assume that the robot starts with a complete (but possibly not correctly labeled) map, the ability to follow a path through the map, a starting point but not an end-point on the map, and a test to determine whether the goal has been reached. We are interested in whether the robot reaches the desired destination, and if so, whether the path taken is the one described by the instructor—since both testing and training maps contain loops, a destination can always be reached in any of several ways, but our primary interest is in how successfully the robot is following directions.

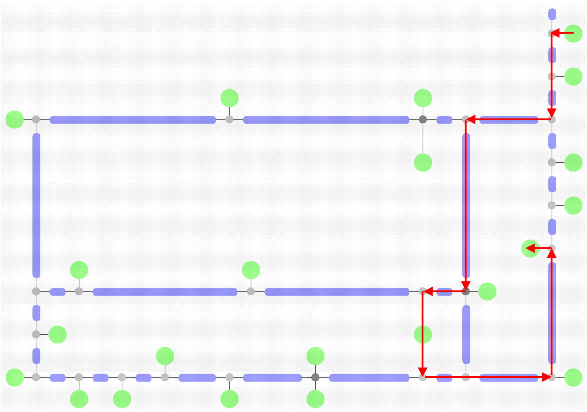


Fig. 8. The training map, produced from the Allen Center computer science building. The red line shows one of the paths the route instructors were asked to describe in the training data.

Training and testing maps were generated from Voronoi graph maps of two buildings, the Allen Center and Intel Research (pictured in Figure 8 and Figure 3, respectively.) These maps were constructed from real laser range-finder data collected by a SICK scanner mounted on a small Pioneer robot base. Because the path description language is unambiguous with respect to movement through a map, it is feasible to generate the formally described component of training examples. Since paths can be generated randomly, the process of collecting training data is primarily constrained by our ability to obtain natural human descriptions of paths.

To obtain training data, five random paths were generated on the map. on the training map; the paths were not always the shortest possible route between start and end points. For this work, we obtained a collection of 33 sets of route instructions describing five paths from eight volunteers. Three of the five paths were non-optimal. The paths varied in length from 4-9 instruction segments and 15-20 nodes traversed. In order to increase the size of the training set, the training data was analyzed and phrases corresponding to individual actions (such as "turn left") were extracted and randomly combined to describe other paths through the map, producing a much larger synthetic data set, into which words were randomly introduced to provide noise. (This allows for the learning of word gap models, although those models would in this case be strictly random.) Some examples of (noiseless) route instructions produced in this way are:

Leave the room and go left, go past the next junction, and then take the second right.

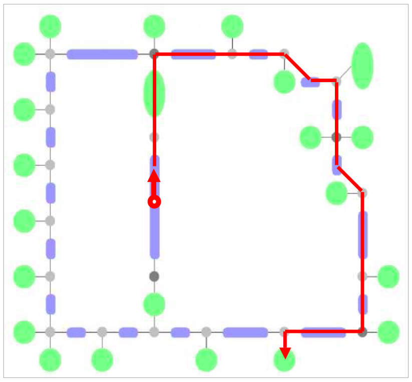
Go straight through the second junction and take a right, pass the next junction, and go into the room ahead.

As noted in Section II-A2, neither landmarks nor context are considered in this work. The former do not appear in the natural-language training data collected. Because volunteers were shown only the final topological graph-map when giving instructions, no additional landmarks were described. However, context-based instructions, which are not representable in the current path description language, were used in some route instructions.

We tested on a corpus of an additional five paths provided by four volunteers over the Intel building testing map, providing 20 sets of route instructions.[2] The testing map was not seen during the training phase. We asked additional human volunteers to follow the directions provided in order to evaluate their quality, and found that 70% of them were followed successfully (fourteen). This is consistent with results reported by others [11], [21].

We found no cases in which our system successfully followed instructions that human evaluators could not follow; this is a likely result of training data which contains very few errors. A larger body of less-perfect training data would probably allow the translation step to provide incorrect parses with low probabilities, which would eventually be tried. Of the fourteen human-followable direction sets, our system was able to follow ten successfully, or 71% (50% of the total), getting to the correct destination along the specific route described by the route director. For a perfect map, the route selected to explore first was the correct one in all cases except those that involved counting (e.g., "Go through two interesctions,"), which entail taking an arbitrary number of steps before the conditions described are met, and sometimes have very strong parse probability differences based on the specific cases encountered in training. Given the nature of the statistical machine translation system, it seems very likely that these results would improve with additional training data and less reliance on synthetic data.

As erroneous map labeling increases, so does the likelihood of exploring an incorrect path initially. For example, if a four-way intersection is incorrectly identified as being a three-way intersection, then a robot instructed to "take the second left" might pass through that intersection rather than turning, preferring the parse that matched the (incorrect) map labels. Because the robot backtracks when its explorations fail, this error is recoverable; the correct parse will be explored and the goal will be reached, although not as efficiently. This behavior—exploring the "best" (i.e., most likely) interpretation and backtracking if the goal is not reached—mimics human direction following. We artificially introduce a random map-labeling error of 90%, which is actually higher than that produced by the Voronoi Random Field approach.



Go past one junction, turn right, take the second right, take the third right, and enter the room on the left.

Fig. 9. One of the routes used for testing, with human-provided directions that could be followed successfully. This set of navigation instructions demonstrates the system's learned capability to follow directions which include counting information (e.g., "take the third right").

Our system successfully generalizes from a few examples to unfamiliar cases, making it able to handle not only input with terms that were not previously encountered, but also cases that require counting (such as in Figure 9) or an understanding of semantic grouping, e.g., the similar uses of the terms "right" and "left." The training data provided to the system for different types of actions is necessarily incomplete; for example, in our tests, we successfully handle commands such as "Take the third left," which requires passing through an unknown number of intersections and rooms or hallways of unknown types. The training data provided examples of a few possible ways of stating the command, with different

[2] Because there was no overlap among people providing directions for training and testing, we asked the testing group to use approximately the same style of giving directions; because direction-giving is known to employ a fairly limited vocabulary, a larger body of training data would make this step unnecessary.

lengths, and novel parses of the command are generated. We demonstrate learning of higher-level semantic groupings such as generalization between the concepts of right and left. The concept of "left" was learned automatically from examples; no information on the *meaning* of left, as a motion, was explicitly encoded in the map or in the path description. From the fact that the two terms appear in otherwise identical examples some of the time, the system was able to transfer learning between left and right in cases where examples of only on or the other was provided, thereby reducing the amount of necessary training data.

IV. RELATED WORK

Navigation is a critical and widely-studied task in mobile robotics, and following natural-language instructions is a key component of natural, multi-modal human/robot interaction [19]. There has been substantial work on mapping and localization [20], segmenting and describing a map from sensor data [7], [8], and navigating through such an environment [15], [25]. Our work fits into the broader class of *grounded language acquisition*, in which language is learned from some situated context, usually by learning over a corpus of parallel language and context data [17], [13]. Learning to follow natural-language directions has been explored in a wide variety of tasks, including scene description [16] and controlling the grasping behaviors of a robotic hand [9], interpreting written software and game manuals [2], and generating commentaries of simulated robocup soccer games [4]. Natural language-directed navigation is the task most closely related to the work presented in this paper.

[11] studied the inference requirements and language used for natural-language navigational instructions in complex simulated environments, as well as providing experimental evidence of the poor quality of instructions provided by human direction-givers. The underlying model of the world used in these trials is more complex than that available in our limited-sensor model, and the instructions are annotated with a semantic grammar based on the semantically rich environment. Because we are using less rich map data collected by laser sensors, we are constrained to solving a similar problem with much less data. As well, we rely entirely on learning from examples, with no human supplied heuristics for managing missing or implies knowledge.

Our approach is similar to [18] with respect to the representation of the path description layer and map layer, but we do not rely on labeled training data, and our natural language segmentation is not map-based. In contrast to [6], the formal representation extracted from instructions defines a path, rather than a set of action and goal representations defined over a hand-annotated map. We share with [21] the goal of probabilistic handling of uncertain environments and noisy input, as well as reasoning over the entire trajectory. However, we rely entirely on an automatically segmented and labeled map with no information about objects encountered or object co-occurence. Most importantly, however, our semantic parser is not limited to nouns and explicit direction terms, as it is trained over more general natural language.

This work would not have been possible without access to WASP, a grounded language acquisition system that can be trained over a parellel corpus of natural language and formal representations in a wide variety of domains.[3] WASP provides SCFG-based statistical parsing and generation capabilities [22], [23]. Additional related work on parsing is described in Section II-B. The simplicity and lack of constraints on our formal language mean that the core SMT approach taken by WASP becomes unmanageable for sequences of instructions; the additional work required to make the problem feasible is described in Section II-C.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have demonstrated a novel system that allows robots to learn, entirely from examples, how to use statistical machine translation to follow natural language directions in a tractable way. We have implemented a version of this system and demonstrated that it shows promise for route instruction following. No information was provided about the alignment of natural language concepts and path descriptions; learning was entirely based on example pairs of instructions and map traces, with no pre-programmed concept of the meaning of natural language phrases such as "room," "intersection," or "turn."

The search mechanism we describe feasibly combines management of uncertainty in parsing and in map labeling; this makes our approach robust to imperfect labeling techniques, meaning it is possible to demonstrate a system that goes all the way from NLP to robot-built maps. Even though our experiments were performed without using a real robot, all the information used by our system was extracted from data collected by a real robot. Our results show approximately a 70% success rate in following instructions given by humans through an entirely novel map, including demonstrated ability to interpret difficult concepts such as counting up to some condition.

A. *FUTURE WORK*

The experimental results demonstrate the potential of our approach. However, there are several improvements in progress which will make human direction-giving easier and more natural. Our tests were performed in a previously-unseen map, suggesting that our training approach does lead to learning of the semantic intent of English route instructions. However, our current system uses a complete map of the environment to perform route-finding. Since there are cases when such a map may not be available *a priori*, we intend to extend our current work to enable online parsing—that is, the robot interprets the directions as it moves through an unknown environment. We also plan to extend the current approach to learn to label maps based on training data; in this way, we will learn a topological map representation that is most suitable

[3] Retrieved from http://www.cs.utexas.edu/∼ml/wasp.

as a target for the natural language examples. In future, we also intend to perform more extensive experiments with actual robots in various environments.

Additionally, the version of this system for which we have experimental results does not handle natural language descriptions that relate to objects or the appearance of certain areas in an environment (i.e., landmarks), which are an important aspect of human navigation [21]. It also does not take context into account; the grammar expresses only information about rooms the robot traverses, and not surrounding spaces, which makes certain commonly-occurring utterances (such as "go to the end of the hall" or "pass the hallway on your left") impossible to express. Neither of these restrictions are inherent limitations of our approach, but rather are a result of the underlying map representation and path description grammar. We are currently extending the Voronoi representation to a more fine-grained spatial resolution along with the ability to represent landmark and object locations in the map, and a more complex version of the grammar has been designed which allows for formal path descriptions which use landmarks in the form of identifiable node types as well as contextual information. We are confident that our approach will be able to parse more general descriptions using such a representation.

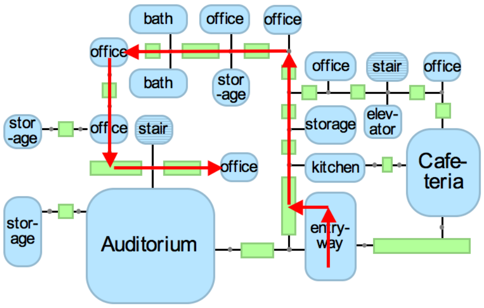


Fig. 10. An example of a map that incorporates labeled nodes, which will allow contributors to describe paths in terms of landmarks seen and passed through. Contributors will be asked to describe the path shown in red through the map.

Because these changes will allow more complex directions to be learned and grounded in a more complex map, we expect to need a larger set of training data to fully cover the richer language allowed. We are pursuing gathering much larger training and test sets using Amazon's Mechanical Turk tool, which will allow us to explore the effectiveness and tractability of this scheme for unconstrained natural language input from a large number of contributors.

VI. ACKNOWLEDGMENTS

The authors gratefully acknowledge the attendees of TCEP 16 for assistance obtaining sample route instructions and discussions of how to best present navigation routes.

REFERENCES

[1] A. V. Aho and J. D. Ullman, *The Theory of Parsing, Translation, and Compiling*. Prentice Hall Professional Technical Reference, 1972.

[2] S. R. K. Branavan, H. Chen, L. Zettlemoyer, and R. Barzilay, "Reinforcement learning for mapping instructions to actions," in *Proc. of the Joint Conf. of the 47th Annual Meeting of the ACL and the 4th Int'l Joint Conference on Natural Language Processing*. Suntec, Singapore: Association for Computational Linguistics, August 2009, pp. 82–90.

[3] P. F. Brown, J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin, "A statistical approach to machine translation," *Comput. Linguist.*, vol. 16, no. 2, pp. 79–85, 1990.

[4] D. Chen and R. Mooney, "Learning to sportscast: a test of grounded language acquisition," in *ICML 2008: Proc. of the 25th international conference on Machine learning*. Helsinki, Finland: ACM, 2008, pp. 128–135.

[5] D. Chiang. (2006) An introduction to synchronous grammars. [Online]. Available: http://www.isi.edu/chiang/papers/synchtut.pdf

[6] J. Dzifcak, M. Scheutz, C. Baral, and P. Schermerhorn, "What to do and how to do it: Translating natural language directives into temporal and dynamic logic representation for goal management and action execution," in *Proc. of the 2009 IEEE Int'l Conf. on Robotics and Automation (ICRA '09)*, Kobe, Japan, May 2009.

[7] S. Friedman, H. Pasula, and D. Fox, "Voronoi random fields: Extracting topological structure of indoor environments via place labeling." in *IJCAI*, M. M. Veloso, Ed., 2007, pp. 2109–2114.

[8] J.-S. Gutmann, M. Fukuchi, and M. Fujita, "3d perception and environment map generation for humanoid robot navigation," *Int. J. Rob. Res.*, vol. 27, no. 10, pp. 1117–1134, 2008.

[9] K.-y. Hsiao, S. Tellex, S. Vosoughi, R. Kubat, and D. Roy, "Object schemas for grounding language in a responsive robot," *Connection Science*, vol. 20, no. 4, pp. 253–276, 2008.

[10] A. Lopez, "Statistical machine translation," *ACM Comput. Surv.*, vol. 40, no. 3, pp. 1–49, 2008.

[11] M. Macmahon, B. Stankiewicz, and B. Kuipers, "Walk the talk: Connecting language, knowledge, action in route instructions," in *In Proc. of the Nat. Conf. on Artificial Intelligence (AAAI)*, 2006, pp. 1475–1482.

[12] E. Martins, M. Pascoal, and J. Santos, "A new improvement for a k shortest paths algorithm," *Investigação Operational*, 2001.

[13] R. J. Mooney, "Learning to connect language and perception," in *Proc. of the Twenty-Third AAAI Conf. on Artificial Intelligence, AAAI 2008*, D. Fox and C. P. Gomes, Eds. Chicago, Illinois: AAAI Press, July 2008, pp. 1598–1601.

[14] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[15] N. D. Ratliff, J. A. Bagnell, and M. A. Zinkevich, "Maximum margin planning," in *In Proc. of the 23rd Int'l Conf. on Machine Learning (ICML06)*, 2006.

[16] D. Roy, "Learning visually-grounded words and syntax for a scene description task," *Computer Speech and Language*, 2002.

[17] D. Roy, "Semiotic schemas: a framework for grounding language in action and perception," *Artificial Intelligence*, vol. 167, no. 1-2, pp. 170–205, 2005.

[18] N. Shimizu and A. Haas, "Learning to Follow Navigational Route Instructions," in *Int'l Joint Conf. on Artificial Intelligence (IJCAI)*, 2009.

[19] M. Skubic, D. Perzanowski, S. Blisard, A. Schultz, W. Adams, M. Bugajska, and D. Brock, "Spatial language for human-robot dialogs," *IEEE Transactions on Systems, Man, and Cybernetics, Part C, Special Issue on Human-Robot Interaction*, vol. 34, no. 2, pp. 154–167, May 2001.

[20] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, September 2005, iSBN 0-262-20162-3.

[21] Y. Wei, E. Brunskill, T. Kollar, and N. Roy, "Where to go: Interpreting natural directions using global inference," in *Int'l Conf. on Robotics and Automation (ICRA)*, 2009.

[22] Y. W. Wong, "Learning for semantic parsing and natural language generation using statistical machine translation techniques," Ph.D. dissertation, Univ. of Texas at Austin, August 2007.

[23] Y. W. Wong and R. J. Mooney, "Learning for semantic parsing with statistical machine translation," in *Proc. of the main conference on Human Language Technology Conf. of the North American Chapter of the Association of Computational Linguistics*. Association for Computational Linguistics, 2006, pp. 439–446.

[24] J. Y. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. 712–716, 1971.

[25] B. Ziebart, A. Maas, A. Dey, and J. D. Bagnell, "Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior," in *UBICOMP: Ubiquitious Computation*, 2008.