Where to Go: Interpreting Natural Directions Using Global Inference

Yuan Wei, Emma Brunskill, Thomas Kollar, Nicholas Roy

***Abstract*— An important component of human-robot interaction is that people need to be able to instruct robots to move to other locations using naturally given directions. When giving directions, people often make mistakes such as labelling errors (e.g., left vs. right) and errors of omission (skipping important decision points in a sequence). Furthermore, people often use multiple levels of granularity in specifying directions, referring to locations using single object landmarks, multiple landmarks in a given location, or identifying large regions as a single location. The challenge is to identify the correct path to a destination from a sequence of noisy, possibly erroneous directions. In our work we cast this problem as probabilistic inference: given a set of directions, an agent should automatically find the path with the geometry and physical appearance to maximize the likelihood of those directions. We use a specific variant of a Markov Random Field (MRF) to represent our model, and gather multi-granularity representation information using existing large tagged datasets. On a dataset of route directions collected in a large third floor university building, we found that our algorithm correctly inferred the true final destination in 47 out of the 55 cases successfully followed by humans volunteers. These results suggest that our algorithm is performing well relative to human users. In the future this work will be included in a broader system for autonomously constructing environmental representations that support natural human-robot interaction for direction giving.**

I. INTRODUCTION

As robots become part of daily life, one essential capability they must have is to be able to interpret and follow human directions in natural human environments. However, people are notoriously poor at giving directions. Firstly, the directions may be noisy or incorrect; people confuse left and right, distance estimates are frequently wrong, and instructions for important decision points may be missing. Furthermore, and more importantly, people do not always share the same perception or representation of an environment. People will often refer to aspects of the environment at different levels of granularity: one person might say "go past the sofa" whereas another might say "walk past the living room." Successful direction giving requires recognizing that both of these observations are consistent references to the same location. This requires more than using an ontology to recognize different class levels of an object, such as realizing "futon" is related to "sofa." Rather it requires knowledge of what scenes tend to have similar objects and labels, such as a kitchen is likely to include a sink and a refrigerator.

We address the problem of determining the correct path through a known environment from a sequence of noisy and ambiguous directions. We cast this as a probabilistic inference problem: given a set of directions an agent must infer the most likely hidden sequence of physical regions corresponding to the given directions. In the current work we assume that the map is segmented in advance into a set of regions, such as by using a SLAM algorithm to construct a hybrid map (see e.g. [13]) or by hand. Our long term goal is to have robots autonomously construct appropriate representations of the environment that they then use to reliably and correctly infer human level directions.

In addressing this challenge, we make two contributions. Firstly, we show that a specific variant of a Markov Random Field (MRF) is a better inference model than existing inference techniques for compensating for errors in the directions. We argue that using probabilistic inference creates a robust and flexible approach: flexibility comes from allowing users to use the representation most natural to them, and robustness comes from using a MRF algorithm to infer the most likely complete path. In large environments where there may be considerable ambiguity and more than one instance of a particular landmark or area, such as schools or businesses with large numbers of offices, bathrooms and kitchens, reasoning about the full trajectory instead of making local decisions may be particularly helpful.

Our second contribution is to use a specific MRF model that allows us to handle directions (or observations) provided at multiple levels of granularity, such as referring to a region by an abstract name (terms such as "kitchen" or "office"), or by a specific object contained in the region (terms such as "microwave" or "computer"). We learn a multi-granularity model from an existing, large, tagged dataset that allows us to infer relationships between known concepts and novel keywords in the directions. The specific dataset we use is Flickr, a dataset of photographs which are tagged with labels by users. By analyzing tags, we can automatically compute shared probabilities between different types of objects, such as the probability of a "microwave" label when a "kitchen" label is also present. These relationships could be achieved by hand labeling a large corpus; however in addition to being time consuming, this has the the drawback that objects not present in the original labeled corpus can cause problems when applying these learned groupings to new environments.

The paper commences with a discussion of related work in Section II. We then present our approach in Section III, starting with some background on probabilistic inference of time-series data, followed by details of our specific algorithm. In Sections IV and VI we evaluate our approach and then conclude.

The authors are with the Computer Science and Artificial Intelligence Laboratory at the Massachusetts Institute of Technology, Cambridge MA, USA. weiy@mit.edu,emma@csail.mit.edu,(tkollar,nickroy)@mit.edu.

II. Related Work

Within the robotics community there has been recent interest in utilizing the structure of the environment when interacting with humans, and a number of researchers have understood the need to characterize space by a hierarchy of elements contained in each place [1], [2], [3]. For example, an office area is likely to entail a desk and a computer. This previous work utilizes semantic networks that have been created by hand, but there is no sense of uncertainty in their spatial models of how *likely* objects are to appear with others. By utilizing hand-constructed semantic representations, researchers have enabled their robots to communicate with people in a meaningful although limited way, and do not consider the use of directions through the environment.

There has, however, been a significant amount of work on how people give natural-language directions. Michon and Denis [4] found that pedestrians perceive landmarks as a useful part of route directions: the authors concluded that when people refer to landmarks they are attempting to provide an abstract topological map, and use landmarks to guide people through difficult or uncertain parts of that environment. In [5], Stoffel constructs a geometric model from which he takes into account topological relationships, visibility within areas, and the generation of route descriptions. In his model, he considers a number of spatial relationships, but does not use uncertainty or landmarks when generating the route descriptions. MacMahon et a. [6] created a system for automatically following natural language route directions. The focus of this system was to infer implicit actions from linguistic conditional phrases with no information about the environment topology. The authors presented results in a virtual maze-like environment with landmarks such as butterflies. The authors' algorithm performed well relative to humans: their approach is mostly orthogonal to the one presented in this paper, and it is possible that by combining some aspects of this system with our own could result in a strong system. One of the conclusions of their study was that landmarks are incredibly important for navigation.

In [7], the authors learned to extract spatial relationships from grid maps. These relationships are given only at a local level. Directions such as "robot, go to the pillar" are parsed from natural language and turned into logical expressions.

Gribble et al. [8] described a robotic wheelchair based upon the Spatial Semantic Hierarchy: their system could potentially follow directions over an extended period, but the authors did not evaluate their assertions. Similarly, Muller et. al [9] described directing a semi-autonomous wheelchair through an environment, where commands take the form of "enter right door." However, the authors did not describe how they dealt with arbitrary landmarks or uncertainty in the locations. Both approaches appear to make hard decisions on directions based on known landmarks and the spatial directions, in contrast to our approach which computes the globally most likely path.

III. Markov Random Fields

We now briefly introduce Markov Random Fields (MRFs) which are the graphical models we use to represent the relationship between physical locations and verbal directions. In the next section we will describe in more detail how we can use MRF inference to find the most likely sequence of states for a given set of observations as a method to find the most likely path of physical locations corresponding to a user's spoken directions.

MRFs are undirected graphical models in which related variables are linked with an edge in order to convey a dependency between the two variables x_i and x_j. This dependency is encoded by a feature function $\phi(x_i, x_j)$. Our MRF consists of a discrete set of states S and a discrete set of observations Z. At each time t the world transitions to a new state s_t and yields a new observation of the new state, z_t. We assume each observation variable z_t is only connected with the state value at the same time step, s_t, and the feature function relating the two is the same for all times t:

$$\phi(z_t, s_t) = \phi(z_t|s_t) = \phi(z|s) \quad \forall t.$$

In other words, there is an observation model that relates observations to states, and this model is fixed for all time.

There will also be interdependencies between the state variables themselves. We assume that our model follows a N-th order Markov process, where N will be defined below. This means that state variable s_t will be conditionally dependent on $s_{t-N}, s_{t-N+1}, \ldots, s_{t-1}$. This dependency will be encoded by a feature function $\phi(s_t|s_{t-1}, \ldots, s_{t-N})$. The value of this feature function for a particular set of state values is proportional to the probability of transitioning to state $s_t = v$ given a particular set of past values for the prior N states. Note that this is only proportional to this distribution since the feature functions between the states are not normalized, ($\sum_{s_t \in S} \phi(s_t|s_{t-1}, \ldots, s_{t-N}) \neq 1$). The full model is shown in Figure 1.

Our MRF is closely related to a hidden Markov model (HMM). The relation between observations and states is similar to the relationship encoded in HMMs; as in HMMs, states in our model are hidden and information about them is only provided through the observations emitted. However, our feature functions are not normalized to sum to 1, unlike the feature functions in HMMs used to define the transition and observation probability tables. The absence of normalization in a MRF allows us to avoid a specific bias during HMM inference of favoring states with a smaller number of potential next states in their transition model (the "label bias" problem [10]). The importance of avoiding this bias in direction inference will be discussed in later sections.

MRFs can be used to address a range of important questions, including what is the most likely state sequence $s_1, s_2, \ldots s_T$ given a series of observations $z_1, z_2, \ldots, z_T$,

$$\underset{s_1,\ldots,s_T}{\operatorname{argmax}} p(s_1, \ldots, s_T|z_1, \ldots, z_T). \tag{1}$$

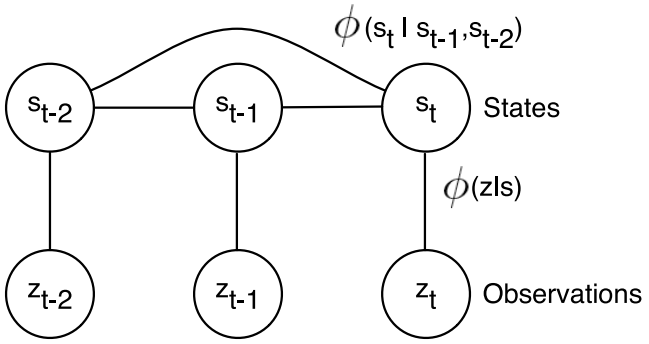

Fig. 1. Specialized Markov Random Field variant used in our approach.

Using Bayes rule this is proportional to

$$\underset{s_1,\ldots,s_T}{\operatorname{argmax}} \phi(z_1,\ldots,z_T|s_1,\ldots,s_T)\phi(s_1,\ldots,s_T).$$

Using the conditional dependencies expressed in our model we can re-express Equation 1 as:

$$\underset{s_1,\ldots,s_T}{\operatorname{argmax}} p(s_1)\phi(z_1|s_1)\prod_{t=2}^{T}\phi(z_t,|s_t)\phi(s_t,s_{t-1},\ldots,s_{t-N})$$

This final expression can be evaluated efficiently using a Viterbi-style algorithm [11].

A. Direction interpretation as inference

We will use our specific MRF to perform direction inference. In our model, states represent the physical regions of an environment. For example, Figure 2 shows the third floor of a building, segmented into a non-overlapping set of contiguous physical regions. Each region is associated with a set of objects that are present in that region: in the current work object labeling is done by hand but in the future we intend this to be performed with an object recognition algorithm. The observations are keywords occurring in a set of directions, such as sofa, kitchen, or monitor. The objective is to compute the most likely sequence of physical regions, given a set of observations (directions). Evaluating the likelihood of sequences requires that we specify the correct transition and observation model probabilities for our problem.

1) Transition Probabilities: In our approach the physical connectivity of the space helps define the feature function encoding the dependencies between prior and future states. In particular, the feature function is zero for all next states $s_t = R_i$ which are not physically adjacent to the prior state region $s_{t-1} = R_j$:

$$\begin{aligned}\phi(s_t = R_i|s_{t-1} = R_j,\ldots,s_{t-N}) &= 1 \text{ if } Adjacent(R_i, R_j)\\ &= 0 \; otherwise\end{aligned}$$

where the $Adjacent$ function is true if it is possible to directly transition between regions R_i and R_j and false otherwise. For example, in Figure 2 regions R1 and R3 are adjacent. A region is also considered to be adjacent to itself. In addition, in a set of directions it is not expected

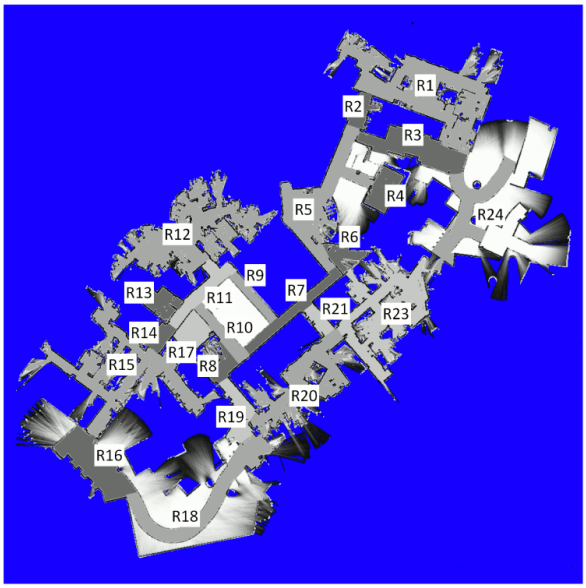

Fig. 2. Stata center segmented into regions. Adjacent regions are shaded differently to highlight the region boundaries.

that the agent will backtrack to previously visited regions. In our initial experiments we only used first order feature functions for the state transitions, and we found that this caused the agent to occasionally return to earlier visited regions, or oscillate, particularly in similar places where the observation model could not uniquely identify the area. Our N-th order feature functions for the state transitions allow us to expressly prohibit this, by setting the feature function value to zero for revisiting states earlier than the state s_{t-2}

$$\phi(s_t = R_i|\exists \tilde{t} \in t-N,\ldots,t-3 \; s_{\tilde{t}} = R_i) = 0.$$

We set N according to the length of the given direction set. Finally, in order to ensure that in absence of any observations, all paths of equal length have the same probability, the feature function values themselves are set to 1 for all allowed transitions. For example, starting in a region connected to five others, the feature function for the next state taking on any of the five next potential region values would be equal to 1. If the first region had instead been connected to two other regions, the feature function value for each of those regions would also have been 1. In this way, transitions through highly connected regions do not receive a lower likelihood than transitions through regions with low connectivity. Note that this is a key distinction from Hidden Markov models which would require the feature function value to sum to 1 over all the next potential regions for any current region, which results in highly connected regions having a lower probability of transitioning to other regions compared to regions that are only adjacent to a few regions.

2) Observation Probabilities: we could pre-compute the model for a complete set of observation probabilities for all observations (aka directions) we expect to receive. However, this pre-computation will inevitably lead to failures when someone gives directions using novel vocabulary. Instead,

we represent regions using a fixed set of abstract types, but we compute the observation model online from the nouns in the parsed directions. These keywords could refer to specific objects o such as 'monitor" or "microwave" or types of regions y, such as "kitchen" or "office." In order to perform inference we need to compute the probability that each map region generated a particular observation, $p(z_i|R_j)$ for all regions j and keywords i. We will assume that each region R_j is associated with a list of objects that were detected in that region $d_{j1}, d_{j2}, \ldots, d_{jD}$. We assume that these object detections are not perfect: that there is a probability θ_{fp1} for a particular object o_1 that we get a false positive, and a probability θ_{fn1} that our object detector fails to detect when there really is an object o_1 in the region. Given this, the probability that there is an object o_1 in region R_j given a detection d_{j1} is

$$p(o_1|d_{j1}, R_j) = p(o_1|d_{j1}) = 1 - \theta_{fp1},$$

namely, the true positive rate of the object detector.

If z_i instead refers to a region type, such as "kitchen" (K), then we use the object detections found in the region R_j to infer the probability that R_j is a kitchen:

$$p(z = K|R_j, d_{j1}, d_{j2}, \ldots, d_{jD}) = p(z = K|d_{j1}, d_{j2}, \ldots, d_{jD})$$

Applying Bayes rule (using K to represent $z = K$) we get

$$\begin{aligned} p(K|d_{j1}, \ldots, d_{jD}) &= \frac{p(d_{j1}, d_{j2}, \ldots, d_{jD}|K)p(K)}{p(d_{j1}, d_{j2}, \ldots, d_{jD})} \\ &= \sum_{\mathcal{O}} \frac{p(d_{j1}, d_{j2}, \ldots, d_{jD}, O|K)p(K)}{p(d_{j1}, d_{j2}, \ldots, d_{jD})} \end{aligned}$$

where O is a particular set of objects present in a region and. Here we are introducing and summing over possible object sets $\mathcal{O}$. We assume that object detections are only dependent on the objects present in the environment (and not the environment type), so we can re-express this as

$$p(K|d_{j1}, \ldots, d_{jD}) = \sum_{\mathcal{O}} \frac{p(d_{j1}, d_{j2}, \ldots, d_{jD}|O)p(O|K)p(K)}{p(d_{j1}, d_{j2}, \ldots, d_{jD})}$$

We next make the simplifying assumption that the probability of each object is independent conditioned on the region type (as in a naive Bayes model), and that each object detection depends on whether or not that particular object is present

$$\begin{aligned} p(K|d_{j1}, \ldots, d_{jD}) &= \sum_{\mathcal{O}} \frac{p(K)\prod_{l=1}^{D} p(d_{jl}|o_l)p(o_l|K)}{p(d_{j1}, d_{j2}, \ldots, d_{jD})} \\ &\propto \sum_{\mathcal{O}} p(K)\prod_{l=1}^{D} p(d_{jl}|o_l)p(o_l|K). \quad (2) \end{aligned}$$

This sum should be over all possible object sets O: if there are N_O objects in the world, there would be 2^{N_O} potential object sets, corresponding to the possibility that each object is or is not truly present in a particular region. For a large number of objects this is intractable: instead we approximate this sum by considering only objects that were detected in a particular region. This effectively means that we consider false detections but not missed object detections.

To make this concrete, consider the case of when there is only 1 possible object in the world (o_1) and we have detected this object (d_{j1}) in region R_j. Then Equation 2 becomes

$$\begin{aligned} p(K|d_{j1}) &\propto p(K)[p(d_{j1}|\neg o_1)p(\neg o_1|K) + p(d_{j1}|o_1)p(o_1|K)] \\ &= p(K)\theta_{fp1}p(\neg o_1|K) + (1 - \theta_{fp1})p(o_1|K). \end{aligned}$$

In other words, the likelihood that region R_j is a kitchen, given that object 1 was detected there is proportional to the probability that either there is or is not truly object 1 in that region, and the associated probabilities related to that.

At a high level, this allows us to create a model that is more robust to the probability that our detections are incorrect. In the case that we have a perfect object detector, this model simplifies as expected.

In order to compute Equation 2 for each region, we must be able to evaluate the probability of an object being present or not present in a particular region type ($p(o|z = K)$ and $p(\neg o|z = K)$). We do this by using a Flickr image dataset. We constructed our dataset by first using WordNet to find all hyponyms for environmental areas (such as hallway, office, etc): this produced approximately 2000 words. Flickr was queried with each of those terms, and around 500 images were downloaded for each term, along with all the associated tags for those 500 images. Given this set of images and tags, we performed simple counting to compute the probabilities

$$\begin{aligned} p(Object|RegionType) &= \frac{p(Object, RegionType)}{p(RegionType)} \\ &= \frac{N_{TagOR}/N_I}{N_{TagR}/N_I} \\ &= \frac{N_{TagOR}}{N_{TagR}} \end{aligned}$$

where N_I is the total number of images in the set, N_{TagR} is the number of images with a tag of $RegionType$ and N_{TagOR} is the number of images with tags of both $RegionType$ and $Object$. The benefit of using Flickr is two-fold: it is an existing labeled dataset, and it is a very large set of images, labeled by a huge number of users. Therefore we expect the probabilities of particular objects being associated with particular region types to be more reliable than if we were to hand label a small set of regions and use these to compute the model probabilities. Here we set the probability of a false detection θ_{fp} manually to be 90% for all object categories: soon we intend to use an automatic object detector and will use its associated false positive rate.

In summary, given a set of observations, we first parse the directions and extract a set of keywords. We then use MRF inference to compute the probability that each keyword corresponds to each region given the keywords, the transition probabilities, and observation probabilities. We then extract the most likely sequence of regions, with two additional modifications. First we constrain the search for the most likely region sequence to start at the known first location in the sequence: we anticipate that our approach will be

Algorithm 1 Algorithm

1: **Input:** English directions D, Map M segmented into regions R, list of objects detected in each region O, and transition model representing connectivity between the regions $\phi(s_t|s_{t-D}, \ldots, s_{t-1})$
2: Parse sentence and extract keywords $d_1, d_2, \ldots, d_N$ from D ("Leave the office and turn right into the hallway" goes to d_1="Office", d_2="Right", d_3="Hallway")
3: Compute observation model for each extracted keyword which is a noun given each of the map regions R. $(\phi(d_1 ='' Office''|R_1), \ldots \phi(d_1 ='' Office''|R_N))$
4: Run the Viterbi algorithm on the keywords using the transition model and observation model.
5: Return the most likely sequence of regions output from Viterbi.

Fig. 3. One of the images taken of region R12 in the dataset. The objects in a region were used to compute the probability of a region being a particular type of area, such as a kitchen or office.

used in settings where a human is giving instructions to a robot that is present in the room with the human to some other region, and so it is reasonable to assume the start location is known. Second, our current transition model is very simple and does not include information about the agent's orientation, or directional information such as "turn right." However, whenever an instruction such as "right" or "left" is encountered, it is often an indication that the agent is about to change physical regions. Therefore, whenever such a keyword is encountered, the transition probability of a self transition is set to zero, to force a transition to an adjacent region.

An overview of our approach is presented in Algorithm 1.

IV. Evaluation

In order to evaluate our algorithm we gathered directions from 11 volunteers on a set of 10 possible pairs of regions, resulting in 80 total sets of directions. Each volunteer was given the segmented map shown in Figure 2 and asked to write, in whatever way was natural to the person, directions from the start region to the end region, with the goal of communicating this route to another person that does not have a map. User volunteers were familiar with the space, and were encouraged to review the area before writing directions if he or she was unsure of how to give directions between each pair of regions.

	# correct destination / Human performance	# correct destination / Total samples
Humans	55/55 (100%)	55/80 (68.8%)
Algorithm	47/55 (85%)	47/80 (58.75%)
Random guessing	4.4/55 (8%)	4.4/80 (7.8%)

TABLE I
Results on direction accuracy

From these user directions, keywords were extracted by hand, and the algorithm presented in the prior section was run to extract the most likely sequence of regions given the set of observations. Figure 4 gives an example of one of the routes that volunteers were asked to write down directions for, as well as the volunteer directions given, extracted keywords and most likely state sequence output.

In order to fairly evaluate the performance of our algorithm, it was important to first ascertain how good the human directions were. To estimate this we tried giving each of the directions to a different volunteer. In each test, the set of directions for a particular route was read out loud by one of the paper authors as the volunteer tried to follow those directions. If the volunteer thought he/she was lost the trial is finished and the directions were classified as wrong/insufficient. If the volunteer ended up at the wrong destination the directions were also classified as wrong or insufficient. If the volunteer finished at the correct destination the directions were classified as correct. In some scenarios people got confused and thought if they did not already have a very good knowledge of the environment they would have been lost. These direction sets were also classified as wrong. Note that some volunteers had an advantage over generic users, since some volunteers had already given a set of directions for the routes they were tested on. Therefore we expect our evaluation of the average percentage of time humans could follow other humans' directions to be potentially an overestimate of the general case.

Our subjects could correctly infer the final destination region from someone else's directions in 55 examples (on average 68.75% of the time), and the MRF correctly inferred the destination in 47 examples. Though both results leave significant room for improvement, our automated approach compares favorably with human performance. It also indicates that this environment is challenging. Note that random guessing would result in a correct answer only 7.8% of the time, since there are 18 regions in the environment: this would yield an expected number of 4.4 correct answers. These results are displayed in Table I.

Despite its generally encouraging performance (compared to humans), there were some routes that our algorithm performed very poorly on, such as navigating from $R16$ to $R12$. This was a short path from the elevators to another office area, but the quickest path involves going through $R15$, which is a sprawling office bracketed by two sets of

Route : R08 to R04
Directions: "Head down the **hall**way with the open area on your **left**, and **railing** on your **left**. At the end of the **hall**way take a **left**, and head through the open area with the **computers** on your **right**, and then head into the **conference room** across the **bridge** on your **right**."
Parsed keyword sequence:

1	*hall*	7	*computer*
2	*left*	8	*right*
3	*balcony*	9	*conferenceroom*
4	*left*	10	*bridge*
5	*hall*	11	*right*
6	*left*	12	*conferenceroom*

Recognized Path:

1	R08 (lounge)	5	R04 (conference room)
2	R10 (hall)	6	R05 (office)
3	R07 (hall)	7	R04 (conference room)
4	R05 (office)		

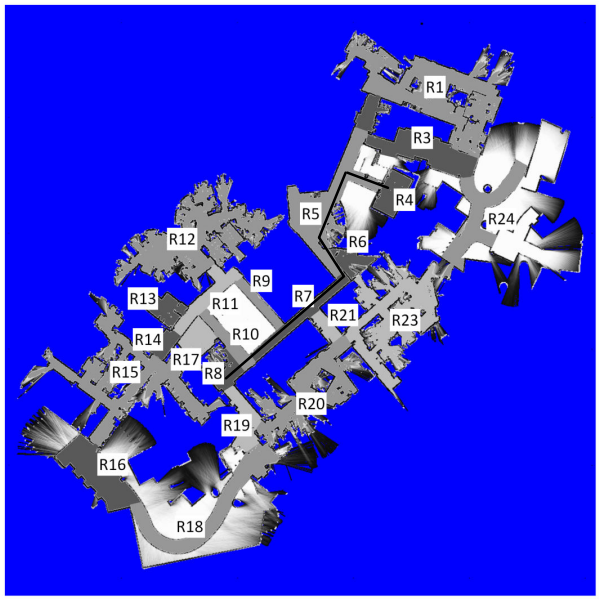

Fig. 4. Sample route, directions given by one volunteer for this route, extracted direction keywords provided as input to the algorithm, and the output best region sequence of the algorithm.

glass doors. This area is quite confusing for humans, and the directions given for this area often involved some extra redundant observations and lots of additional comments. For example, one set of directions given were:

```
1. With your back to the elevators, head
through the glass doors on your left.
2. Follow the hallway past the biolab,
and through the doors at the end, and
all the way down the hallway.
3. At the end, with the open area on
your right, take a left and head into
the office area.
```

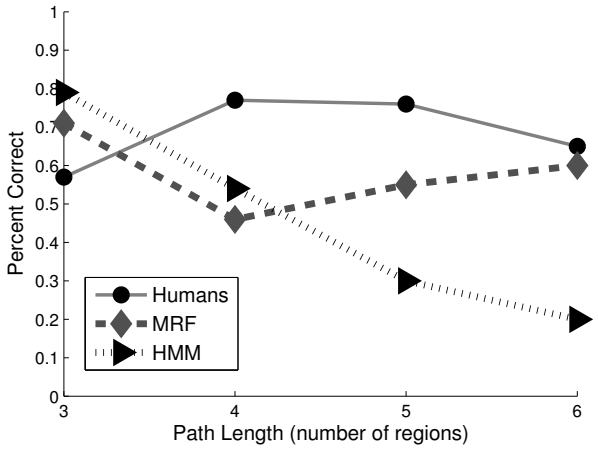

Fig. 5. Percentage of correct directions as a function of route path length.

. The extracted keywords for this sequence were *'elevators', 'doors', 'null', 'left', 'hall', 'doors', 'hall', 'right', 'null', 'left', 'office', 'office'*. The most likely sequence of regions found by the algorithm was *'R16 elevators', 'R15 office', 'R11 hall', 'R09 hall', 'R07 hall', 'R06 office', 'R06 office'*, which is far past the desired end trajectory. Similar results were found for this route from some of the other directions. One interesting thing about this set of directions is that information about side regions is given multiple times, such as "past the biolab" and "with the open area on your right." We will discuss this further in the future work section.

However, it is exciting that in some cases the algorithm performed much better than humans. For example, for the route between $R6$ to $R2$ only 2 out of 6 people got the final destination correctly, but the algorithm got it correct in all 6 cases. Figure 5 shows the results as a function of path length. Using a MRF does significantly better than a HMM as expected. Recall that the major difference between our MRF model and the HMM model is whether the transition probabilities are normalized. This normalization means that highly connected regions will have a much lower transition probability than regions with few adjacent regions. Therefore, particularly as the path length gets longer, the HMM will tend to favor paths with regions with few connections, in order to maximize the probability. Our MRF model does not have this disadvantage, and performs better, particularly as the path length increases. Overall our MRF algorithm shows encouraging results and with further improvements, it may be possible to be competitive with human performance across all path lengths, though further testing is needed.

V. Future Work

The work presented makes up a first encouraging initial step towards our longer term goal of a completely autonomous system for direction following. We are currently working on the following extensions to the algorithm:

- *Skipped observations*. In giving directions, people sometimes skip observations of regions by using higher level action instructions. For example, "Leave the office

area and walk with the wall on your right until you see the kitchen" could be used to specify how to walk from $R12$ to $R2$, but there are no observations given of the regions $R9$, $R6$, or $R5$ that the follower must traverse on the way to kitchen $R2$. We are currently extending the algorithm to include potential skips in the observation sequence ("null" observations in the sequence where a state transition occured). We are taking a string matching alignment approach to this problem, and preliminary results on hand inserting skips in the observation direction sequence are promising.

- *Alternate descriptions*. In our original model we assume that people provide "feed forward" descriptions, in which the only regions described are along the chosen route. However, in our instruction set people sometimes refer to regions or objects that are adjacent to the region that a person or agent should be in presently, such as "with the open area to your right." People also sometimes use negative information, such as "if you have reached the drug store you have gone too far." Incorporating both types of information is likely to significantly improve our algorithm's performance. A simple way to incorporate observations of side regions would be to modify the observation model $p(z|s)$ for each region to include observations associated with the region itself and with adjacent regions. In addition, in certain tasks the speaker may backtrack to past regions, though we expect that to be unlikely in the types of problems we are interested in (since our current focus is not on environment "tours").
- *Improved parsing*. Our current parser automatically extracts the nouns and simple direction terms (such as "right" or "left") in the order they are presented. People however use rich linguistic structure to encode their directions, such as "turn into the office, after going past the kitchen and the bathroom," which would be currently parsed so the kitchen and bathroom appear to come after the office, instead of before.
- *Automatic map and region generation*. In the longer term our goal is to use a robot to automatically build a region based representation of the environment, perhaps by using past hybrid metric-topological map building algorithms (e.g. [12], [13]). During this map building the robot will also take photos (such as shown in Figure 3) and automatically detect what objects are present in the photo, and associate these objects with the appropriate map region. We have already made progress on this challenge but our object detection method did not yet have high enough recognition rates to be used in the current presented work. After improving this we will be able to use these regions and object detections in our direction inference algorithm. We are also interested in examining the impact of the chosen map segmentation on the ability of the robot to infer the correct path: a range of segmentations may enable successful inference.
- *Interactive direction giving*. We are also currently pursuing work where the original set of directions provided is only the start of a dialogue between a human and a robot: the robot can then ask clarification questions in order to ascertain the correct path or destination. We are taking a decision theoretic approach to this problem, in which asking additional questions involves a cost of the potential annoyance factor to the human.

VI. Conclusion

We have posed the problem of direction following as a probabilistic inference problem, framing the objective as inferring the hidden sequence of physical regions refered to by a given set of human directions. Our model correctly computed the true destination at a rate of 85% compared to humans. We are encouraged by these results and think our future work may make our approach even more competitive.

VII. Acknowledgements

Y. Wei was supported by AFOSR under the Agile Robotics project, contract number 7000038334. E. Brunskill and N. Roy were supported by the NSF Division of Information and Intelligent Systems under grant # 0546467. E. Brunskill also received support from a Hugh Hampton Young Memorial Fund Fellowship. T. Kollar was supported by the Office of Naval Research under MURI N00014-07-1-0749. We appreciate the support of these organizations and the kind assistance of the study volunteers.

References

[1] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, F. J. A. Madrigal, and J. Gonzalez, "Multi-hierarchical semantic maps for mobile robotics," *IROS*, 2005.

[2] K. B. Bruckner, U. Frese, K. Luttich, C. Mandel, T. Mossakowski, and R. Ross, "Specification of an Ontology for Route Graphs," *Spatial Cognition IV: Reasoning, Action, Interaction. International Conference Spatial Cognition*, pp. 390–412, 2004.

[3] H. Zender, M. O. Mozos, P. Jensfelt, G. J. M. Kruijff, and W. Burgard, "Conceptual spatial representations for indoor mobile robots." *Robotics and Autonomous Systems*, vol. 56, pp. 493–502, 2008.

[4] P. E. Michon and M. Denis, "When and why are visual landmarks used in giving directions," *Spatial Information Theory. Lecture Notes in Computer Science*, vol. 2205, pp. 292–305, 2001.

[5] E.-P. Stoffel, B. Lorenz, and H. Ohlbach, "Towards a semantic spatial model for pedestrian indoor navigation," *Advances in Conceptual Modeling Foundations and Applications*, pp. 328–337, 2007.

[6] B. K. M. MacMahon, B. Stankiewicz, "Walk the talk: Connecting language, knowledge, and action in route instructions," in *AAAI*, 2006.

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

[8] W. S. Gribble, R. L. Browning, M. Hewett, E. Remolina, and B. J. Kuipers, "Integrating Vision and Spatial Reasoning for Assistive Navigation," *Lecture Notes in Computer Science*, pp. 179–193, 1998.

[9] R. Muller, T. Rofer, A. Lankenau, A. Musto, K. Stein, and A. Eisenkolb, "Coarse Qualitative Descriptions in Robot Navigation," *Lecture Notes In Computer Science*, vol. 1849, pp. 265–276, 2000.

[10] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data", in *ICML*, 2001.

[11] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[12] M. Bosse, P. M. Newman, J. J. Leonard, and S. Teller, "Slam in large-scale cyclic environments using the atlas framework," *International Journal of Robotics Research*, vol. 23, no. 12, pp. 1113–1139, 2005.

[13] E. Brunskill, T. Kollar, and N. Roy, "Topological Mapping Using Spectral Clustering and Classification," in *IROS*, 2007.