

# Learning to Give Route Directions from Human Demonstrations

Stefan Oßwald

Henrik Kretzschmar

Wolfram Burgard

Cyrill Stachniss

**Abstract**—For several applications, robots and other computer systems must provide route descriptions to humans. These descriptions should be natural and intuitive for the human users. In this paper, we present an algorithm that learns how to provide good route descriptions from a corpus of human-written directions. Using inverse reinforcement learning, our algorithm learns how to select the information for the description depending on the context of the route segment. The algorithm then uses the learned policy to generate directions that imitate the style of the descriptions provided by humans, thus taking into account personal as well as cultural preferences and special requirements of the particular user group providing the learning demonstrations. We evaluate our approach in a user study and show that the directions generated by our policy sound similar to human-given directions and substantially more natural than directions provided by commercial web services.

## I. INTRODUCTION

Providing and following route directions is a task we face on a daily basis, for example, when asking staff members how to get to the right aisle in a library or shopping center, or when moving in an unknown city using instructions provided by a navigation device. When driving cars, we already rely on web services such as Google Maps [1] and satellite navigation systems that provide turn-by-turn instructions. For some applications, however, giving turn-by-turn instructions is suboptimal, for example, in indoor navigation where global positioning is not available. In such applications, computer systems should be able to give spoken directions of the complete route in advance. This is relevant for information kiosks in public buildings, tour guide robots in museums, or navigation devices for visually impaired persons.

Existing commercial web services typically follow rigid patterns for giving directions, usually referring to the metrical distance to the forthcoming turning and the name of the street that the user has to take, e.g., “Turn left after 267 m onto Market Street.” While these descriptions meet the requirement of guiding us to our target, they do not resemble the directions a human would give. Instead of providing precise metric information, humans usually refer to salient landmarks along the route [2], trade off the amount of information that the recipient requires against the memory load caused by long descriptions, and take into account personal and cultural preferences when giving route directions [3].

The goal of this work is to develop a system that gives route directions in spoken form that humans can naturally

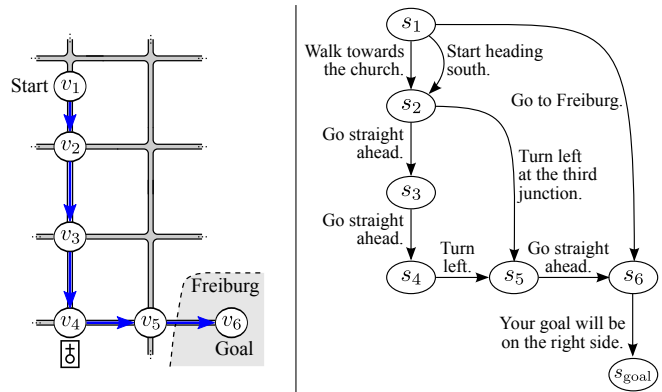


Fig. 1. The problem of giving route directions as a reinforcement learning problem. Left: The route to be described (blue). Right: The corresponding Markov decision process encodes all valid descriptions of the route.

and intuitively understand. We propose an approach that learns how to generate route descriptions from a corpus of human-generated descriptions. In this way, the user can train our system to use a particular style of instructions that is appropriate for the use case at hand. One of the key challenges is to choose the right amount and type of information that a route description should contain. What constitutes a “good” description depends on the user and the situation. For example, street names are hardly of use for visually impaired or blind recipients. The person giving the directions has to trade off the amount of information to give to the recipient. More detailed descriptions may help the user to resolve ambiguities and recover from errors but on the other hand require more memory and are thus harder to remember. Hund *et al.* [3] found in a user study that cultural backgrounds influence the style of route directions, for example the use of street name and cardinal directions (US) vs. landmarks (The Netherlands). Many other factors influence the decision of which ingredients to choose for composing a route description, such as personal preferences, and the situation.

The contribution of this paper is a novel approach to learn a model of human-like directions for routes. We formulate the problem of giving directions as a Markov decision process. Hence, we assume that humans seek to optimize an unknown reward function when giving directions. We apply maximum entropy inverse reinforcement learning to infer the reward function from a set of directions provided by humans. The reward function is expressed in terms of features that capture relevant properties of the directions. Our experimental evaluation suggests that the directions generated by our approach are significantly more human-like than the directions provided by a baseline method.

All authors are with the Department of Computer Science, University of Freiburg, 79110 Freiburg, Germany. Cyrill Stachniss is also with the University of Bonn, Institute of Geodesy and Geoinformation, 53115 Bonn, Germany. This work has partially been supported by the German Research Foundation (DFG) under grant number EXC 1086 and under contract number SFB/TR-8. Their support is gratefully acknowledged.

## II. RELATED WORK

Studies in psychology, cognition, geo sciences, linguistics, and computer science have investigated the principles underlying the process of giving route descriptions. The work by Allen [4] and Lovelace *et al.* [5] identify common characteristics of good route descriptions. For instance, humans typically give directions in linear order along the projected route. Richter [6] provides an overview over related studies and discusses the underlying cognitive principles. While studies such as [2] point out the importance of including landmarks in route descriptions, there is no clear consensus on how to select appropriate landmarks for including them into route descriptions. Many factors affect the route directions given by humans, such as cultural backgrounds [3], gender [7], and familiarity with the environment [5]. As it is impossible to model all related influences, we instead propose a learning algorithm that adapts the style of giving route directions to the preferences of the particular user group by imitating humans.

Several research groups investigated how natural language processing can be used to understand and follow route directions [8], [9], which can be seen as the inverse problem to generating directions. While their results provide valuable insights, their methods are not directly applicable to the problem of generating new route descriptions.

Look [10] implemented a system for generating overview descriptions consisting of subgoals and neighborhoods along the route that complement and structure turn-by-turn instructions according to cognitive principles. Dale *et al.* [11] manually analyzed references and structures in a corpus of human-written directions and derived a set of rules for generating route descriptions that appear to be more human-like. In our work, we follow a similar approach but attempt to learn the user preferences directly from the human-generated corpus instead of manually inferring a set of rules.

Several algorithms have been proposed for altering the route in order to generate better descriptions, for example, for finding the most reliable path [12], or the simplest instructions [13]. Goeddel *et al.* [14] propose to use a particle filtering technique for maximizing the chances for arriving at the goal despite possible errors the user makes while traveling. In contrast to these approaches, our algorithm learns policies that cater to specific user groups with particular needs and preferences. The generated descriptions are not necessarily optimal with respect to efficiency, reliability, simplicity, or similar metrics, as descriptions given by humans are not optimal according to these metrics either.

Cuayáhuil *et al.* [15] propose a hierarchical reinforcement learning approach for choosing a route and generating suitable directions along that route. The learned policy minimizes the number of instructions necessary to describe the route and the “user confusion” based on hand-crafted reward functions and transition models that consider the user’s familiarity with the environment, the distance between subsequent decision points, and the saliency of the landmarks that the directions refer to. In contrast to that, our approach learns how much information and which pieces of information the route description should

include by imitating descriptions given by humans without the need for estimating parameters of the underlying model, such as the “user confusion” probabilities.

Inverse reinforcement learning techniques have been used to address a variety of imitation learning problems including autonomous helicopter aerobatics [16], learning pedestrian navigation behavior [17], [18], and learning the preferences of cab drivers [19]. In particular, Abbeel and Ng [20] suggest to match features that capture relevant aspects of the behavior that is to be imitated. However, feature matching does not lead to a unique cost function. To resolve this ambiguity, Maximum Entropy Inverse Reinforcement Learning [21] relies on the principle of maximum entropy [22] and, hence, aims to find the policy with the highest entropy subject to feature matching. In this work, we apply Maximum Entropy Inverse Reinforcement Learning to learn a model of “human-like” route descriptions from a set of directions given by humans. Our approach is able to imitate their style of route directions in order to generate route descriptions that appear natural and intuitive to the recipients.

## III. LEARNING TO GIVE DIRECTIONS FOR ROUTES FROM HUMAN DEMONSTRATIONS

The objective of this work is to learn a model of natural-language route directions from human demonstrations. We model the problem of giving directions for routes as a reinforcement learning problem in terms of a Markov decision process. We apply inverse reinforcement learning to elicit the unknown reward function from a set of directions provided by humans. The reward function captures relevant characteristics of the route directions and allows us to apply reinforcement learning to generate directions for new routes that mimic the human demonstrations.

### A. Giving directions as a reinforcement learning problem

A Markov Decision Process (MDP) is given by a set of states  $s \in S$ , a set of actions  $a \in A$ , state transition dynamics in terms of a probability distribution  $P(s' | s, a)$  of the next state  $s'$ , and a reward function  $r : S \times A \rightarrow \mathbb{R}$ . A trajectory  $\tau = \{s_{1:T}, a_{1:T}\}$  is a sequence of states and actions that obeys the state transition dynamics. A policy  $\pi(a | s)$  is given by a probability distribution over actions  $a$  to take when in state  $s$ . The goal of reinforcement learning is to compute a policy  $\pi(a | s)$  that maximizes the expected cumulative reward. In contrast to that, inverse reinforcement learning is the problem of recovering an unknown reward function given a set of demonstrations  $\{\tau_1, \dots, \tau_n\}$ . In this work, we apply inverse reinforcement learning as a reward function cannot be specified easily.

As illustrated in Fig. 1, we model the environment  $m$  as a graph  $G = (V, E)$ , where the nodes  $v \in V$  represent intersections, and the edges  $e \in E$  represent streets. The space of all directions that guide a user through the environment  $m$  from its current location  $v_1$  to a specific destination location  $v_{\text{goal}}$  along a given route is encoded by a deterministic MDP  $M$ . In particular, each action  $a$  of the MDP corresponds to a single instruction that guides the user from node  $v_i$  to node  $v_j$ .

Consequently, each trajectory  $\tau$  in the MDP  $M$  corresponds to valid directions that guide the user all the way from its current location to its target location.

### B. Maximum entropy inverse reinforcement learning

Maximum entropy inverse reinforcement learning (MaxEnt IRL) [21] assumes that the behavior of the agent is governed by a probability distribution of trajectories that is a function of features  $\mathbf{f}_\tau \in \mathbb{R}^n$ . The principle of maximum entropy [22] suggests to select the distribution that least favors any particular outcome while satisfying observed constraints. Accordingly, MaxEnt IRL considers the maximum entropy distribution of trajectories whose expected feature values match the observed feature values  $\tilde{\mathbf{f}}_\tau$ , yielding

$$\tilde{\mathbf{f}}_\tau = \mathbb{E}_{p_\theta}[\mathbf{f}(\tau)]. \quad (1)$$

Hence, the algorithm optimizes the objective function

$$p_\theta^* = \operatorname{argmax}_{p_\theta} H_{p_\theta(\tau)} \text{ such that } \tilde{\mathbf{f}}_\tau = \mathbb{E}_{p_\theta}[\mathbf{f}(\tau)], \quad (2)$$

where  $H(p_\theta)$  is the Shannon entropy of distribution  $p_\theta$ . The resulting distribution  $p_\theta^*(\tau)$  is given by

$$p_\theta^*(\tau) \propto e^{-\theta^T \mathbf{f}(\tau)}, \quad (3)$$

and the gradient with respect to its parameters  $\theta$  is

$$\tilde{\mathbf{f}}_\tau - \mathbb{E}_{p_\theta}[\mathbf{f}(\tau)]. \quad (4)$$

We apply gradient-based optimization to iteratively refine the parameters  $\theta$  to achieve feature matching. To this end, we use Ziebart's efficient dynamic programming algorithm [21] to compute the expected visitation frequency  $D_\theta(s_i, a_i)$  of each state-action pair  $(s_i, a_i) \in \tau$ . The expected feature values  $\mathbb{E}_{p_\theta}[\mathbf{f}(\tau)]$  induced by the current parameters  $\theta$  are then given by

$$\mathbb{E}_{p_\theta}[\mathbf{f}(\tau)] = \sum_{(s_i, a_i) \in \tau} D_\theta(s_i, a_i) \mathbf{f}(s_i, a_i). \quad (5)$$

### C. Features

Inverse reinforcement learning uses feature vectors to characterize each possible solution of the task to be learned. In our scenario, the features represent properties of the route directions such as the length of the description, the landmarks mentioned in the instructions, or the frame of reference used to indicate turn directions.

Tab. I shows all features implemented in our system. They can be grouped into three categories:

a) *Amount of information*: The amount of information that the directions should include depends on the situation in which the user gives the directions to the recipient. For example, if the recipient will receive the directions as a printed text that he can take along while traveling, the descriptions can be more detailed compared to situations in which he has to memorize the whole description. We introduce two features to measure the amount of information on different levels: First, we count the total number of instructions. Second, we define a feature measuring the number of pieces of information within each instruction inspired by Mark's

TABLE I  
FEATURES CHARACTERIZING ROUTE DESCRIPTIONS. SEE SEC. III-C FOR  
DETAILED EXPLANATIONS.

Feature	Description
number of instructions	total number of instructions
number of slots	number of pieces of information an instruction contains (see Sec. III-C)
abstraction level	turn-by-turn instruction, chunked instruction, or destination description
segment length	length of the described route segment in meters
number of intersections	number of intersections described by the instruction
street name references	number of street names mentioned
street name saliency	saliency of the street name if mentioned
cross street references	number of crossing street names mentioned
cross street saliency	saliency of crossing street if mentioned
street category ref.	number of street categories mentioned
cross street cat. ref.	number of crossing street categories
"head towards" ref.	number, saliency and distance of landmarks that indicate the heading, but are not part of the route, e.g. "head towards the ocean"
"head towards" saliency	
"head towards" distance	
landmark reference	number of landmarks mentioned
landmark distance	distance of landmark if mentioned
landmark direction	direction of landmark relative to recipient (back, side, ahead, etc.)
landmark saliency	saliency of landmark based on the number of hits returned from a web search for the landmark's name (see Sec. IV-A for details)
turn anchor	turn at landmark / after metric distance / etc.
metric reference	discretized mileage mentioned, e.g. "turn right after 200 m"
counter reference	counter value mentioned, e.g. "turn right at the second intersection"
reference frame at turns	cardinal (e.g. "turn north"), allocentric (e.g. "turn left"), or using counter (e.g. "take the second exit in the roundabout")
reference frame at start	cardinal, relative to landmark, or using street name (e.g. "Start on Main St.")
"n times" chunking	e.g. "turn right twice"

concept of "slots" [23]. For instance, the instruction "Turn right at the second intersection into Market Street" contains three pieces of information that the recipient has to remember.

b) *Abstraction levels*: Turn-by-turn instructions refer to each decision point individually (e.g. "Turn right"), chunked instructions that direct the recipient across multiple decision points (e.g. "Go straight ahead until you get to the church"), and destination descriptions just give an intermediate goal without specifying how to get there (e.g. "Drive to Freiburg"). We define features that count the frequencies of these abstraction levels as well as features measuring the number of decision points and the length of the route segment covered by an instruction.

c) *Description style*: Personal and cultural preferences also influence the description style, e.g., the person giving the directions can use cardinal directions, allocentric directions, or directions relative to landmarks to indicate in which direction the recipient has to travel. Furthermore, they can choose to give street names, mileages, etc. Therefore, we define features that count the frequencies of these types of information.

For learning how to choose suitable landmarks, we define features of the saliency of the landmarks (see Sec. IV-A), the distance between the landmarks and the decision points, and the directions of the landmarks relative to the user.

TABLE II  
CONTEXTS CHARACTERIZING THE ROUTE SEGMENT AND ITS  
ENVIRONMENT. SEE SEC. III-D FOR DETAILED EXPLANATIONS.

Context	Description
is at start	true if recipient is at start location
is at goal	true if recipient is at goal location
is dead end	true if recipient's position is in a dead end
is turn	true if recipient has to change direction
turn angle	discretized turn angle (hard, right, slight)
navigation complexity	complexity measure of the intersection
landmark present	true if a suitable landmark is nearby
best landmark saliency	highest saliency of nearby landmarks
street name saliency	saliency of the current street name
street category	category of the current street (e.g. foot path)
street category diff.	difference between street category of current and next road segment

#### D. Contexts

The description of a particular route segment typically depends on the environment of the route segment. For example, turning at complex intersections requires more information than simply going straight ahead. To integrate the context information into the learning process, we introduce contexts as functions  $c : A \rightarrow K$  that map properties of the route segment covered by actions  $A$  to discrete, finite sets  $K$  (e.g. the set of discretized turn angles). In contrast to features, contexts only depend on the route and its surroundings, but not on the user's description.

For example, we calculate a complexity measure of each intersection based on the number and geometry of the streets meeting at that intersection, the direction in which the route continues, and the street names of the inbound and outbound segment of the route. Similarly, we define discretized measures for the turn direction angle, for the street category ranging from foot paths to motorways, and for the presence of landmarks near the decision point.

In line with Ziebart's approach for context-aware inverse reinforcement learning [19], we consider features  $f^k$  that are only active in a particular context, i.e.,

$$\forall k \in K : f^k(a) := \begin{cases} f(a) & \text{if } c(a) = k \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

To determine which features depend on which contexts, we define for each feature  $f$  the probability distribution  $P(f)$  as the histogram of feature values observed during the demonstrations. We then compute for each feature and each context the mutual information (information gain)

$$MI(f; c) = H(f) - H(f | c), \quad (7)$$

where  $H(\cdot)$  denotes the Shannon entropy, and  $P(c)$  is the distribution of context values in the training data. The mutual information  $MI(f; c)$  measures how informative the context is for determining the distribution of feature values. Following the approach commonly used in decision tree learning [24], we combine the features and contexts yielding the highest information gain. For instance, the "is turn" context provides a high information gain for the feature "street name references", as humans mention street names more often when turning than when going straight ahead.

## IV. EXPERIMENTAL EVALUATION

We conducted a two-part user study for evaluating our approach. In the first part, we collected a corpus of directions for a set of given routes from the participants. Using this corpus, we learned a reward function, which we then used to generate descriptions for a set of test routes. In the second part of the experiment, we presented human-written and computer-generated directions to other participants and asked them to rate how natural the descriptions sound to them.

#### A. Acquiring training data

In order to acquire training data for learning a route description policy, we asked 13 participants in a web-based survey to describe up to three routes within Freiburg. All participants were locals and fluent in English. We provided an unmodified, interactive OpenStreetMap map to the participants and asked them to describe the route marked on the map. The instructions introducing the experiment stated that the recipient of the description was a non-local business partner who had some rudimentary knowledge about the city. The recipient would print out the directions and take them along while walking, but would not have a map at his disposal.

In this experiment setup, we collected a corpus of 28 descriptions for ten routes ranging from 0.6 km to 2.9 km in both urban and downtown environments.

For each route, our system generated natural-language instructions corresponding to the route segments based on OpenStreetMap data. All generated descriptions are valid in the sense that they correctly describe the navigation actions the recipient has to execute in order to reach the target.

In order to estimate the saliency of landmarks, we use a two-step metric. First, the algorithm counts the frequency of each category of landmarks in the human-written descriptions, yielding a saliency value for each landmark category. We then weight the individual saliency of each landmark within its group. For estimating landmark saliency, commonly used metrics rely on the visual appearance and semantic attraction of the landmark (e.g. [25]). Unfortunately, publicly available data sources such as OpenStreetMap do not provide the information these metrics require. Hence, our algorithm resorts to querying a web search engine to get a rough saliency estimate. For landmarks with unique proper names (e.g. *Eiffel Tower*), the algorithm searches for the location and exact name of the landmark (e.g. *Paris "Eiffel Tower"*) and estimates the saliency of the landmark based on the relative number of hits. We argue that landmarks that are often mentioned on the web are likely to be better known and more salient. For unnamed landmarks and landmarks without unique names, such as branches of banks or fast-food chains, we estimated the individual saliency based on the density of landmarks of the same type in the vicinity, arguing that landmarks are only valuable if they are distinguishable and not ambiguous. If the map contained enough information on the shape and physical appearance of the landmarks, then the distinctiveness and the visibility of the landmarks from the user's view point could also be incorporated into the saliency measure.

TABLE III  
EXAMPLE INSTRUCTIONS FOR A GIVEN ROUTE

<b>Human-provided description</b>
Start going through the Katharinenstraße. When you get to a huge intersection, turn left onto the big street. Go straight until you come to the trail of the tram. Turn right there. Walk along the trail of the tram, then at the Bertoldsbrunnen you have to turn left onto the Salzstraße. Walk up to the Augustinermuseum, where you have to turn right and enter the Augustinerplatz.
<b>Generated by our algorithm</b>
On Katharinenstraße start heading towards the Altstadt. Go along the street until you encounter a large intersection. Turn left onto the Friedrichstraße, and pass the Vapiano on your right. Go ahead until you come to the tram track. Turn right there onto the Kaiser-Joseph-Straße. Turn left after the Drogerie Müller. Turn right at the second possibility and enter the Augustinerplatz. You are at your target when you get there.
<b>Google Maps description (similar result for Bing and MapQuest)</b>
Head north on Katharinenstraße toward Rheinstraße. Turn right onto Rheinstraße. Turn right onto Merianstraße. Continue onto Rathausplatz. Continue onto Universitätsstraße. Turn left onto Bertoldstraße. Continue onto Salzstraße. Turn right onto Augustinerplatz.

As natural language processing is beyond the scope of this work, we manually matched each description in the collected corpus to the closest description generated by the MDP corresponding to the route. This step implicitly corrects obvious errors in the human-provided descriptions, such as left-right confusions or miscounted intersections, as the generating model always produces correct instructions. We also replaced descriptions of the appearance of landmarks with the name of the respective landmark as we only consider which landmarks to include in the directions, but not how to describe them in words.<sup>1</sup>

Based on the corpus of route descriptions, our algorithm learned a weight vector. To generate a description for a novel route, the algorithm generates the corresponding MDP, computes a policy for the MDP using Ziebart’s algorithm, and samples a path through the MDP according to the learned policy to generate a route description.

### B. Experimental setup

The goal of our work is to imitate the way humans give directions. Hence, we evaluated how well the learned policy imitates the directions given by humans in a second web-based experiment. We provided the participants with a route marked on an interactive OpenStreetMap map together with either a human-written description gathered in the first experiment, a description generated automatically according to the learned policy, or a description generated by one of three popular web services (Google Maps, Bing, and MapQuest).

For the computer-generated descriptions, we used the same routes as in the first experiment to allow for direct comparison with the human-written descriptions, as well as four novel routes. We asked the participants to answer the question “How natural does this description sound to you?” by dragging a

<sup>1</sup>Example: rewrite “you reach a square with a statue of a rider in the middle” as “you reach the Bertoldsbrunnen fountain.” In future work, an additional step could be introduced in the text generation procedure that substitutes the names of the landmarks back to descriptions generated from additional sources, such as the Wikipedia entry of the landmark or information from tourist guide books.

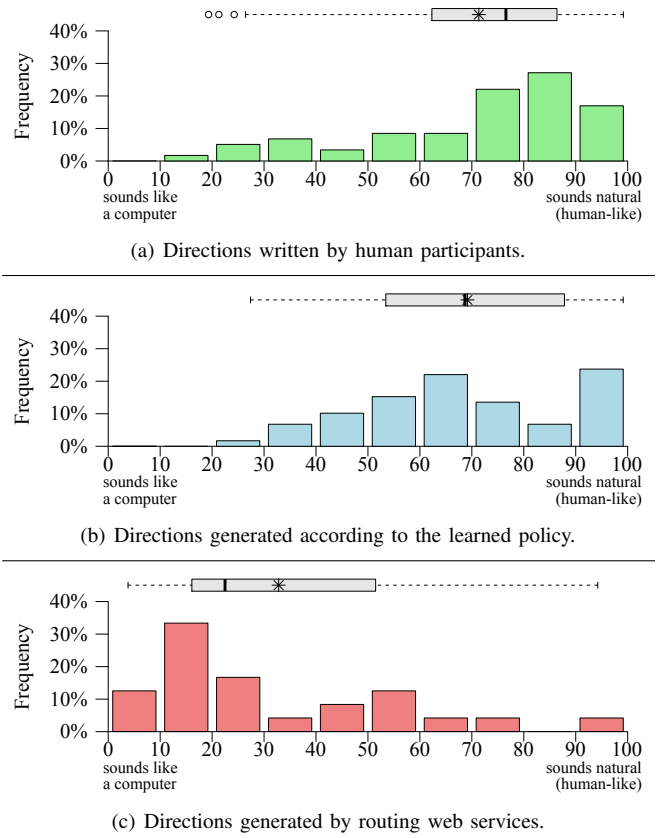


Fig. 2. Results of the second user study. The values on the horizontal axis represent the user ratings of the directions from 0 (“sounds like a computer”) to 100 (“sounds natural (human-like)”). The box plots show the minimum, lower quartile  $Q_1$ , median (thick line), mean (asterisk), upper quartile  $Q_3$ , and maximum, and outliers (circles). Data points are considered to be outliers if they are outside the 1.5 interquartile range  $[Q_1 - 1.5(Q_3 - Q_1), Q_3 + 1.5(Q_3 - Q_1)]$ .  $N = 54$  for (a) and (b),  $N = 22$  for (c).

continuous slider between “sounds like a computer” and “sounds natural (human-like).” Each subject rated twelve descriptions in random order, each description corresponding to a different route. The subjects were allowed to go back to previous routes to revise their answer.

We processed the human-written descriptions according to the manual annotations that the algorithm used for learning the policy. We furthermore fixed typos and spelling errors. As a result of that, the regenerated descriptions have the same structure and contain the same information as the original input, but use the same formatting as the computer-generated instructions. Tab. III provides example descriptions for a given route.

### C. How human-like are the directions generated by our approach?

Fig. 2 shows the distribution of the ratings that the participants of the second experiment assigned to the routes and the corresponding box plots. The participants generally rated the descriptions generated by our approach to sound more natural than the descriptions provided by the commercial web services. The difference is statistically highly significant (t-test with 99.9% significance).



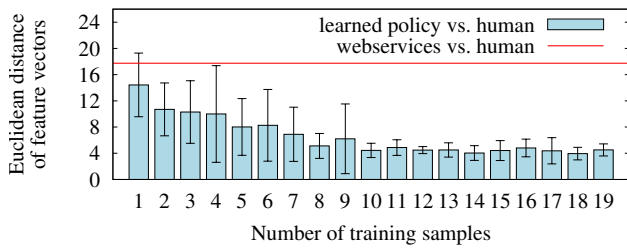


Fig. 3. The difference between the feature vector of the demonstrations and the learned feature expectations quickly converges.

The participants rated the generated descriptions and the human-written descriptions similarly. The mean of the slider values are slightly higher for the human-written descriptions (71.7) compared to the generated descriptions (69.5), but the difference is not statistically significant. Both distributions have high variances, reflecting that there is no clear consensus among the participants what constitutes “natural” instructions.

We furthermore analyzed the number of training instances our system requires. Fig. 3 suggests that a low number of training samples is sufficient to achieve this level of similarity between human and generated directions, as the Euclidean distance between the feature vector of the demonstrations and the feature expectations quickly converges.

The generated descriptions sometimes contain repetitions, for example if two subsequent instructions refer to the same landmark or if the description contains instructions like “Keep going straight ahead” multiple times in a row. According to participants of the second experiment, repetitions make the directions appear less natural. In our framework, however, the Markov property of the MDP prohibits to define a suitable feature for measuring repetitions, as features may only depend on the current action, but not on the previous path leading up to the current state. An additional filtering step after generating the descriptions could address this problem by combining subsequent repetitive instructions to a single instruction that sounds more natural.

## V. CONCLUSIONS

In many applications, robots and other computer systems are required to provide route descriptions to humans. It is desirable that these descriptions are natural and intuitive to human users. In order to generate such natural descriptions, we presented an approach that is based on inverse reinforcement learning and computes a reward function based on human demonstrations. This allows for generating route directions that sound natural to humans by imitating the description style of a particular user group. Our algorithm learns from a corpus of human-written route descriptions which pieces of information are relevant depending on the route and its environment. We carried out a user study to evaluate the quality of the descriptions generated by our approach in terms of how natural they appear to human participants. The results suggest that our learning algorithm produces descriptions that sound similarly natural as human-written descriptions and clearly outperform the descriptions generated by commercial routing web services.

## REFERENCES

- [1] Google Inc. Google Maps. <https://maps.google.com>.
- [2] D. Waller and Y. Lipka, “Landmarks as beacons and associative cues: Their role in route learning,” *Memory & Cognition*, vol. 35, no. 5, 2007.
- [3] A. M. Hund, M. Schmorrow, and M. L. Noordzij, “The impact of culture and recipient perspective on direction giving in the service of wayfinding,” *Journal of Environmental Psychology*, vol. 32, no. 4, pp. 327–336, 2012.
- [4] G. L. Allen, “From knowledge to words to wayfinding: Issues in the production and comprehension of route directions,” in *Proc. of the Int. Conf. on Spatial Information Theory (COSIT)*. London, UK: Springer-Verlag, 1997, pp. 363–372.
- [5] K. L. Lovelace, M. Hegarty, and D. R. Montello, “Elements of good route directions in familiar and unfamiliar environments,” in *Proc. of the Int. Conf. on Spatial Information Theory (COSIT)*. London, UK: Springer-Verlag, 1999, pp. 65–82.
- [6] K.-F. Richter, *Context-specific route directions: generation of cognitively motivated wayfinding instructions*, ser. Dissertationen zur künstlichen Intelligenz. Akademische Verlagsgesellschaft (Aka), 2007.
- [7] S. L. Ward, N. Newcombe, and W. F. Overton, “Turn left at the church, or three miles north: A study of direction giving and sex differences,” *Environment and Behavior*, vol. 18, no. 2, pp. 192–213, 1986.
- [8] M. T. MacMahon, “Following natural language route instructions,” Ph.D. dissertation, Electrical and Computer Engineering Department, University of Texas at Austin, 2007.
- [9] T. Kollar, “Learning to understand spatial language for robotic navigation and mobile manipulation,” Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, USA, 2011.
- [10] G. W. K. Look, “Cognitively-inspired direction giving,” Ph.D. dissertation, Massachusetts Inst. of Technology, Cambridge, USA, 2008.
- [11] R. Dale, S. Geldof, and J.-P. Prost, “Using natural language generation in automatic route description,” *Journal of Research and Practice in Information Technology*, vol. 36, no. 3, pp. 23–39, 2005.
- [12] S. Haque, L. Kulik, and A. Klippel, “Algorithms for reliable navigation and wayfinding,” in *Spatial Cognition V Reasoning, Action, Interaction*, ser. Lecture Notes in Computer Science, T. Barkowsky, M. Knauff, G. Ligozat, and D. Montello, Eds. Springer Verlag, 2007, vol. 4387.
- [13] K.-F. Richter and M. Duckham, “Simplest instructions: Finding easy-to-describe routes for navigation,” in *Proc. of the Int. Conf. on Geographic Information Science (GIScience)*. Springer-Verlag, 2008, pp. 274–289.
- [14] R. Goeddel and E. Olson, “DART: A particle-based method for generating easy-to-follow directions,” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Oct. 2012.
- [15] H. Cuayáhuil, N. Dethlefs, L. Frommberger, K.-F. Richter, and J. Bateman, “Generating adaptive route instructions using hierarchical reinforcement learning,” in *Proc. of Spatial Cognition*. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 319–334.
- [16] P. Abbeel, A. Coates, and A. Ng, “Autonomous helicopter aerobatics through apprenticeship learning,” *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1608–1639, 2010.
- [17] B. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. Bagnell, M. Hebert, A. Dey, and S. Srinivasa, “Planning-based prediction for pedestrians,” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009, pp. 3931–3936.
- [18] M. Kuderer, H. Kretschmar, C. Sprunk, and W. Burgard, “Feature-based prediction of trajectories for socially compliant navigation,” in *Proc. of Robotics: Science and Systems*, Sydney, Australia, 2012.
- [19] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Deyvivi, “Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior,” in *Proc. of the Int. Conf. on Ubiquitous Computing (Ubicomp)*, 2008, pp. 322–331.
- [20] P. Abbeel and A. Ng, “Apprenticeship learning via inverse reinforcement learning,” in *Proc. of the Int. Conf. on Machine Learning*, 2004.
- [21] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, “Maximum entropy inverse reinforcement learning,” in *Proc. of the National Conf. on Artificial Intelligence (AAAI)*, vol. 3, 2008, pp. 1433–1438.
- [22] E. T. Jaynes, “Where do we stand on maximum entropy,” *Maximum Entropy Formalism*, pp. 15–118, 1978.
- [23] D. M. Mark, “Automated route selection for navigation,” *Aerospace and Electronic Systems Magazine, IEEE*, vol. 1, no. 9, pp. 2–5, 1986.
- [24] J. Quinlan, “Induction of decision trees,” *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.
- [25] C. Nothegger, S. Winter, and M. Raubal, “Selection of salient features for route directions,” *Spatial Cognition & Computation*, vol. 4, no. 2, pp. 113–136, 2004.