Vision-and-Language Navigation: Interpreting visually-grounded navigation instructions in real environments

Peter Anderson[1] **Qi Wu**[2] **Damien Teney**[2] **Jake Bruce**[3] **Mark Johnson**[4]
Niko Sünderhauf[3] **Ian Reid**[2] **Stephen Gould**[1] **Anton van den Hengel**[2]

[1]Australian National University [2]University of Adelaide
[3]Queensland University of Technology [4]Macquarie University

Abstract

A robot that can carry out a natural-language instruction has been a dream since before the Jetsons cartoon series imagined a life of leisure mediated by a fleet of attentive robot helpers. It is a dream that remains stubbornly distant. However, recent advances in vision and language methods have made incredible progress in closely related areas. To enable and encourage the application of vision and language methods to the problem of interpreting visually-grounded navigation instructions, we present the Matterport3D Simulator – a large-scale reinforcement learning environment based on real imagery. Using this simulator, which can in future support a range of embodied vision and language tasks, we provide the first benchmark dataset for visually-grounded natural language navigation in real buildings – the Room-to-Room (R2R) dataset.

1 Introduction

The idea that we might be able to give general, verbal instructions to a robot and have at least a reasonable probability that it will carry out the required task is one of the long-held goals of robotics, and artificial intelligence (AI). Despite significant progress, there are a number of major technical challenges that need to be overcome before robots will be able to perform general tasks in the real world. One of the primary requirements will be new techniques for linking natural language to vision and action in *unstructured, previously unseen environments*. It is the navigation version of this challenge that we refer to as Vision-and-Language Navigation (VLN).

Although interpreting natural-language instructions has received significant attention previously [8, 9, 14, 22, 24, 28], it is the recent success of recurrent neural network methods for the joint interpretation of images and natural language that motivates the VLN task, and the associated Room-to-Room (R2R) dataset described below. The dataset particularly has been designed to simplify the application of Vision-and-Language methods to what might otherwise seem a distant problem.

Previous approaches to natural language command of robots have often neglected the visual information processing aspect of the problem. Using rendered, rather than real images [5, 18, 30], for example, constrains the set of visible objects to the set of hand-crafted models available to the renderer. This turns the robot's challenging open-set problem of relating real language to real imagery into a far simpler closed-set classification problem. The natural extension of this process is that adopted in works where the images are replaced by a set of labels [9, 28]. Limiting the variation in the imagery inevitably limits the variation in the navigation instructions also. What distinguishes the VLN challenge is that the agent is required to interpret a previously *unseen* natural-language

Visually-Grounded Interaction and Language (ViGIL) workshop, NIPS 2017.

Instruction: Head upstairs and walk past the piano through an archway directly in front. Turn right when the hallway ends at pictures and table. Wait by the moose antlers hanging on the wall.

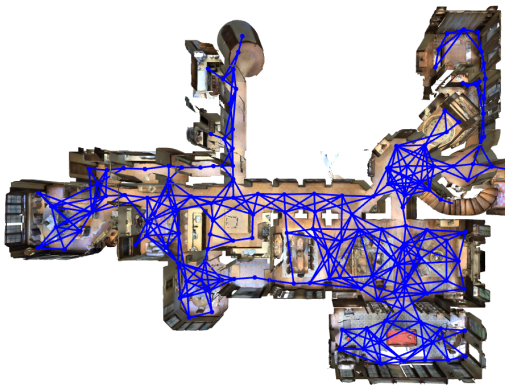

Figure 1: Left: Room-to-Room (R2R) navigation task. We focus on executing natural language navigation instructions in previously unseen real-world buildings. The agent's camera can be rotated freely. Blue discs indicate nearby (discretized) navigation options. Right: Example navigation graph for a partial floor of one building-scale scene in the Matterport3D Simulator. Navigable paths between panoramic viewpoints are illustrated in blue.

navigation command in light of images generated by a previously *unseen* real environment. The task thus more closely models the distinctly open-set nature of the underlying problem.

To enable the reproducible evaluation of VLN methods, we present the Matterport3D Simulator. The simulator is a large-scale interactive reinforcement learning (RL) environment constructed from 10,800 densely-sampled panoramic RGB-D images of 90 real-world building-scale indoor environments [7]. Compared to synthetic RL environments [5, 18, 30], the use of real-world image data preserves visual and linguistic richness, maximizing the potential for trained agents to be transferred to real-world applications.

Based on the Matterport3D Simulator, we collect the Room-to-Room (R2R) dataset containing 21,567 open-vocabulary, crowd-sourced navigation instructions with an average length of 29 words. Each instruction describes a trajectory traversing typically multiple rooms. As illustrated in Figure 1, the associated task requires an agent to follow natural-language instructions to navigate to a goal location in a previously unseen building. For full details refer to our arXiv paper 3. An early release of our code and data for the simulator, dataset and baseline models is available[1].

2 Related work

Navigation and language Within the natural language processing (NLP) community, most existing approaches to the natural language command of robots abstract away the problem of visual perception to a significant degree. This is typically achieved either by assuming that the set of all navigation goals, or objects to be acted upon, has been enumerated, and that each will be identified by label [9, 28], or by operating in severely restricted or simulated environments requiring limited perception [8, 14, 15, 19, 20, 22, 29]. Our work contributes for the first time a navigation benchmark dataset that is both linguistically and visually rich, moving closer to real scenarios while still enabling reproducible evaluations.

Vision and language The development of new benchmark datasets for image captioning [10], visual question answering (VQA) [4, 13] and visual dialog [12] has spurred considerable progress in vision and language understanding. However, although many tasks combining visual and linguistic reasoning have been motivated by their potential robotic applications [4, 12, 17, 21, 27], none of these tasks allow an agent to move or control the camera. Our proposed R2R benchmark fills this gap.

[1] https://github.com/peteanderson80/Matterport3DSimulator

Navigation based simulator Our simulator is similar in spirit to recently proposed 3D RL environments based on game engines, such as ViZDoom [18], DeepMind Lab [5] and THOR [30]. The main advantage of our framework is that all pixel observations come from real images of diverse indoor scenes, making almost every coffee mug, pot-plant and wallpaper texture unique. In seeking to strike a more favorable balance between interactivity and visual realism, we share similar motivations with the much smaller Active Vision Dataset [2], which is also based on densely sampled real-world images.

RL in navigation Chaplot et al. [2017] develop an RL model to execute template-based instructions in the 3D virtual Doom environment, such as 'go to the `[color]`. `[object]`'. Misra et al. [2017] introduce more complex language instructions, while keeping the environment (a virtual Blocks world) fully-observable and visually limited. By releasing the Matterport3D Simulator and R2R dataset, we hope to provide a more realistic partially-observable setting to encourage further research.

3 Matterport3D Simulator

3.1 Matterport3D dataset

Most RGB-D datasets are derived from video sequences; e.g. NYUv2 [25], SUN RGB-D [26] and ScanNet [11]. These datasets typically offer only one or two paths through a scene, making them inadequate for simulating robot motion. In contrast to these datasets, the recently released Matterport3D dataset [7] contains a comprehensive set of panoramic views. To the best of our knowledge it is also the largest currently available RGB-D research dataset.

In detail, the Matterport3D dataset consists of 10,800 panoramic views constructed from 194,400 RGB-D images of 90 building-scale scenes. On average, panoramic viewpoints are distributed throughout the entire walkable floor plan of each scene at an average separation of 2.25m. Each panoramic view is comprised of 18 RGB-D images captured from a single 3D position at the approximate height of a standing person. The dataset also includes globally-aligned, textured 3D meshes that are fully-annotated with class and instance segmentations of both regions and objects.

3.2 Simulator

We extend the Matterport3D dataset by constructing an interactive reinforcement learning (RL) environment for each building-scale scene (the Matterport3D Simulator).

Observations To construct the simulator, we allow an embodied agent to virtually 'move' throughout a scene by adopting poses coinciding with panoramic viewpoints. Agent poses are defined in terms of 3D position $v \in V$, heading $\psi \in [0, 2\pi)$, and camera elevation $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, where V is the set of 3D points associated with panoramic viewpoints in the scene. At each step t, the simulator outputs an RGB image observation o_t corresponding to the agent's first person camera view. Images are generated from perspective projections of precomputed cube-mapped images at each viewpoint. Future extensions to the simulator will also support depth image observations (RGB-D), and additional instrumentation in the form of rendered object class and object instance segmentations (based on the underlying Matterport3D mesh annotations).

Action space The main challenge in implementing the simulator is determining the state-dependent action space. Naturally, we wish to prevent agents from teleporting through walls and floors, or traversing other non-navigable regions of space. Therefore, at each step t the simulator also outputs a set of next step reachable viewpoints $W_{t+1} \subseteq V$. Agents interact with the simulator by selecting a new viewpoint $v_{t+1} \in W_{t+1}$, and nominating camera heading ($\Delta\psi_{t+1}$) and elevation ($\Delta\theta_{t+1}$) adjustments. Actions are deterministic.

To determine W_{t+1}, for each scene the simulator includes a weighted, undirected graph over panoramic viewpoints, $G = \langle V, E \rangle$, such that the presence of an edge signifies a robot-navigable transition between two viewpoints, and the weight of that edge reflects the straight-line distance between them. To construct the graphs, we ray-traced between viewpoints in the Matterport3D scene meshes to detect intervening obstacles. Figure 1 illustrates a partial example of a typical navigation graph. On average each graph contains 117 viewpoints, with an average vertex degree of 4.1. This

compares favorably with grid-world navigation graphs which, due to walls and obstacles, must have an average degree of less than 4. Although agent motion is discretized, many 3D simulators that notionally support continuous motion also use discretized action spaces in practice [30].

The simulator does not define or place restrictions on the agent's goal, reward function, or any additional context (such as natural language navigation instructions). These aspects of the RL environment are task and dataset dependent, for example as described in Section 4.

Implementation Ddetails The Matterport3D Simulator is written in C++ using OpenGL. In addition to the C++ API, Python bindings are also provided, allowing the simulator to be easily used with deep learning frameworks such as Caffe [16] and TensorFlow [1], or within RL platforms such as ParlAI [23] and OpenAI Gym [6].Various configuration options are offered for parameters such as image resolution and field of view. Separate to the simulator, we have also developed a WebGL browser-based visualization library for collecting text annotations of navigation trajectories using Amazon Mechanical Turk, which we will make available.

4 Room-to-Room (R2R) navigation dataset

Task As illustrated in Figure 1, the R2R task requires an embodied agent to follow natural language instructions to navigate from a starting pose to a goal location in the Matterport3D Simulator. Formally, at the beginning of each episode the agent is given as input a natural language instruction $\bar{x} = \langle x_1, x_2, \dots x_L \rangle$, where L is the length of the instruction and x_i is a single word token. The agent observes an initial RGB image o_0, determined by the agent's initial pose comprising a tuple of 3D position, heading and elevation $s_0 = \langle v_0, \psi_0, \theta_0 \rangle$. The agent must execute a sequence of actions $\langle s_0, a_0, s_1, a_1, \dots, s_T, a_T \rangle$, with each action a_t leading to a new pose $s_{t+1} = \langle v_{t+1}, \psi_{t+1}, \theta_{t+1} \rangle$, and generating a new image observation o_{t+1}. The episode ends when the agent selects the special `stop` action, which is augmented to the simulator action space defined in Section 3.2. The task is successfully completed if the sequence of actions leads the agent to an intended goal location v^*.

Data collection To generate navigation data, we use the Matterport3D region annotations to sample start pose s_0 and goal location v^* pairs that are (predominantly) in different rooms. For each pair, we find the shortest path $v_0 : v^*$ in the relevant weighted, undirected navigation graph G, discarding paths that are shorter than 5m, and paths that contain less than four or more than six edges. In total we sample 7,189 paths with an average length of 10m.

For each path, we collect three associated navigation instructions using Amazon Mechanical Turk (AMT). To this end, we provide workers with an interactive 3D WebGL environment depicting the path from the start location to the goal location using colored markers. AMT workers can interact with the trajectory as a 'fly-through', or pan and tilt the camera at any viewpoint along the path for additional context. The full collection interface was the result of several rounds of experimentation. We used only US-based workers, screened according to their performance on previous tasks. Over 400 workers participated in the data collection, contributing around 1,600 hours of annotation time.

R2R dataset analysis In total, we collected 21,567 navigation instructions with an average length of 29 words. This is considerably longer than visual question answering datasets where most questions range from four to ten words [4]. However, given the focused nature of the task, the instruction vocabulary is relatively constrained, consisting of around 3.1k words (approximately 1.2k with five or more mentions).

5 Conclusion

Vision-and-Language Navigation (VLN) is important because it represents a significant step towards capabilities critical for practical robotics. To further the investigation of VLN, in this paper we introduced the Matterport3D Simulator. This simulator achieves a unique and desirable trade-off between reproducibility, interactivity, and visual realism. Leveraging these advantages, we collected the Room-to-Room (R2R) dataset. The R2R dataset is the first dataset to evaluate natural language navigation instruction following, in previously unseen real images at building scale.

Acknowledgments

We would like to thank Matterport for allowing the Matterport3D dataset to be used by the academic community, as well as the Matterport photographers who agreed to have their data included. Collection of the navigation dataset was generously supported by a Facebook ParlAI Research Award. This research is also supported by an Australian Government Research Training Program (RTP) Scholarship, by the Australian Research Council Centre of Excellence for Robotic Vision (project number CE140100016), and by the Australian Government through the Australian Research Council's Discovery Projects funding scheme (project DP160102156).

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

Phil Ammirato, Patrick Poirson, Eunbyung Park, Jana Kosecka, and Alexander C Berg. A dataset for developing and benchmarking active vision. In *ICRA*, 2017.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-Language Navigation: Interpreting visually-grounded navigation instructions in real environments. *arXiv preprint arXiv:1711.07280*, 2017.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual question answering. In *ICCV*, 2015.

Charles Beattie, Joel Z Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, et al. Deepmind lab. *arXiv preprint arXiv:1612.03801*, 2016.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

Devendra Singh Chaplot, Kanthashree Mysore Sathyendra, Rama Kumar Pasumarthi, Dheeraj Rajagopal, and Ruslan Salakhutdinov. Gated-attention architectures for task-oriented language grounding. *arXiv preprint arXiv:1706.07230*, 2017.

David L Chen and Raymond J Mooney. Learning to interpret natural language navigation instructions from observations. In *AAAI*, 2011.

Xinlei Chen, Tsung-Yi Lin Hao Fang, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. Microsoft COCO Captions: Data Collection and Evaluation Server. *arXiv preprint arXiv:1504.00325*, 2015.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, Jose M. F. Moura, Devi Parikh, and Dhruv Batra. Visual dialog. In *CVPR*, 2017.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *CVPR*, 2017.

Sergio Guadarrama, Lorenzo Riano, Dave Golland, Daniel Go, Yangqing Jia, Dan Klein, Pieter Abbeel, Trevor Darrell, et al. Grounding spatial relations for human-robot interaction. In *IROS*, 2013.

Albert S Huang, Stefanie Tellex, Abraham Bachrach, Thomas Kollar, Deb Roy, and Nicholas Roy. Natural language command of an autonomous micro-air vehicle. In *IROS*, 2010.

Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara L. Berg. Referit game: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014.

Michał Kempka, Marek Wydmuch, Grzegorz Runc, Jakub Toczek, and Wojciech Jaśkowski. ViZDoom: A Doom-based AI research platform for visual reinforcement learning. In *IEEE Conference on Computational Intelligence and Games*, 2016.

Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas Roy. Toward understanding natural language directions. In *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*, pages 259–266. IEEE, 2010.

Matt MacMahon, Brian Stankiewicz, and Benjamin Kuipers. Walk the talk: Connecting language, knowledge, and action in route instructions. In *AAAI*, 2006.

Junhua Mao, Huang Jonathan, Alexander Toshev, Oana Camburu, Alan Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016.

Hongyuan Mei, Mohit Bansal, and Matthew R Walter. Listen, attend, and walk: Neural mapping of navigational instructions to action sequences. In *AAAI*, 2016.

Alexander H. Miller, Will Feng, Adam Fisch, Jiasen Lu, Dhruv Batra, Antoine Bordes, Devi Parikh, and Jason Weston. Parlai: A dialog research software platform. *arXiv preprint arXiv:1705.06476*, 2017.

Dipendra K Misra, John Langford, and Yoav Artzi. Mapping instructions and visual observations to actions with reinforcement learning. In *EMNLP*, 2017.

Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A rgb-d scene understanding benchmark suite. In *CVPR*, 2015.

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. MovieQA: Understanding stories in movies through question-answering. In *CVPR*, 2016.

Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R Walter, Ashis Gopal Banerjee, Seth J Teller, and Nicholas Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *AAAI*, 2011.

Adam Vogel and Dan Jurafsky. Learning to follow navigational directions. In *ACL*, 2010.

Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *ICRA*, 2017.

Supplementary Material

Pass the pool and go indoors using the double glass doors. Pass the large table with chairs and turn left and wait by the wine bottles that have grapes by them.

Walk straight through the room and exit out the door on the left. Keep going past the large table and turn left. Walk down the hallway and stop when you reach the 2 entry ways. One in front of you and one to your right. The bar area is to your left.

Enter house through double doors, continue straight across dining room, turn left into bar and stop on the circle on the ground.

Standing in front of the family picture, turn left and walk straight through the bathroom past the tub and mirrors. Go through the doorway and stop when the door to the bathroom is on your right and the door to the closet is to your left.

Walk with the family photo on your right. Continue straight into the bathroom. Walk past the bathtub. Stop in the hall between the bathroom and toilet doorways.

Walk straight passed bathtub and stop with closet on the left and toilet on the right.

Exit the office then turn left and then turn left in the hallway and head down the hallway until you get to a door on your left and go into office 359 then stop.

Go out of the room and take a left. Go into the first room on your left.

Leave the office and take a left. Take the next left at the hallway. Walk down the hall and enter the first office on the left. Stop next to the door to office 359.

Go up the stairs and turn right. Go past the bathroom and stop next to the bed.

Walk all the way up the stairs, and immediately turn right. Pass the bathroom on the left, and enter the bedroom that is right there, and stop there.

Walk up the stairs turn right at the top and walk through the doorway continue straight and stop inside the bedroom.

Figure 2: Examples of navigation instructions from the R2R dataset. For each trajectory we show the view from the agent's starting pose and the three associated navigation instructions.

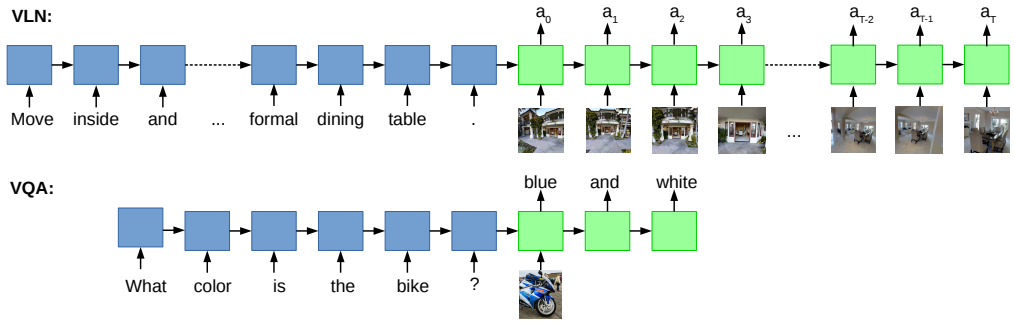

Figure 3: Differences between Vision-and-Language Navigation (VLN) and Visual Question Answering (VQA). Both tasks can be formulated as visually grounded sequence-to-sequence transcoding problems. However, VLN sequences are much longer and, uniquely among vision and language benchmark tasks using real images, the model outputs actions $\langle a_0, a_1, \ldots a_T \rangle$ that manipulate the camera viewpoint.

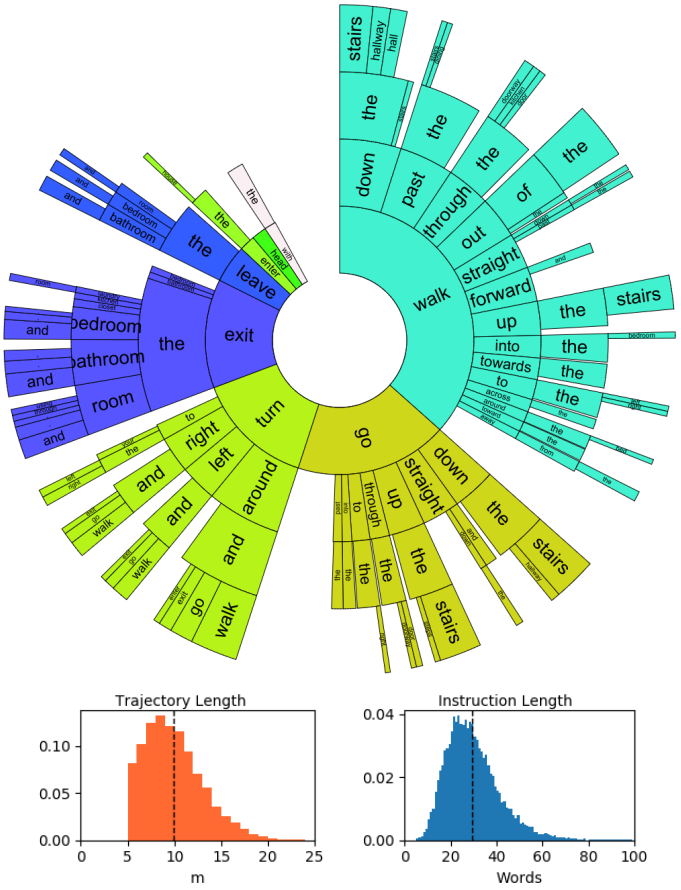

Figure 4: Top: Distribution of navigation instructions based on their first four words. Instructions are read from the center outwards. Arc lengths are proportional to the number of instructions containing each word. White areas represent words with individual contributions too small to show. Bottom: Navigation trajectory lengths and instruction word lengths.

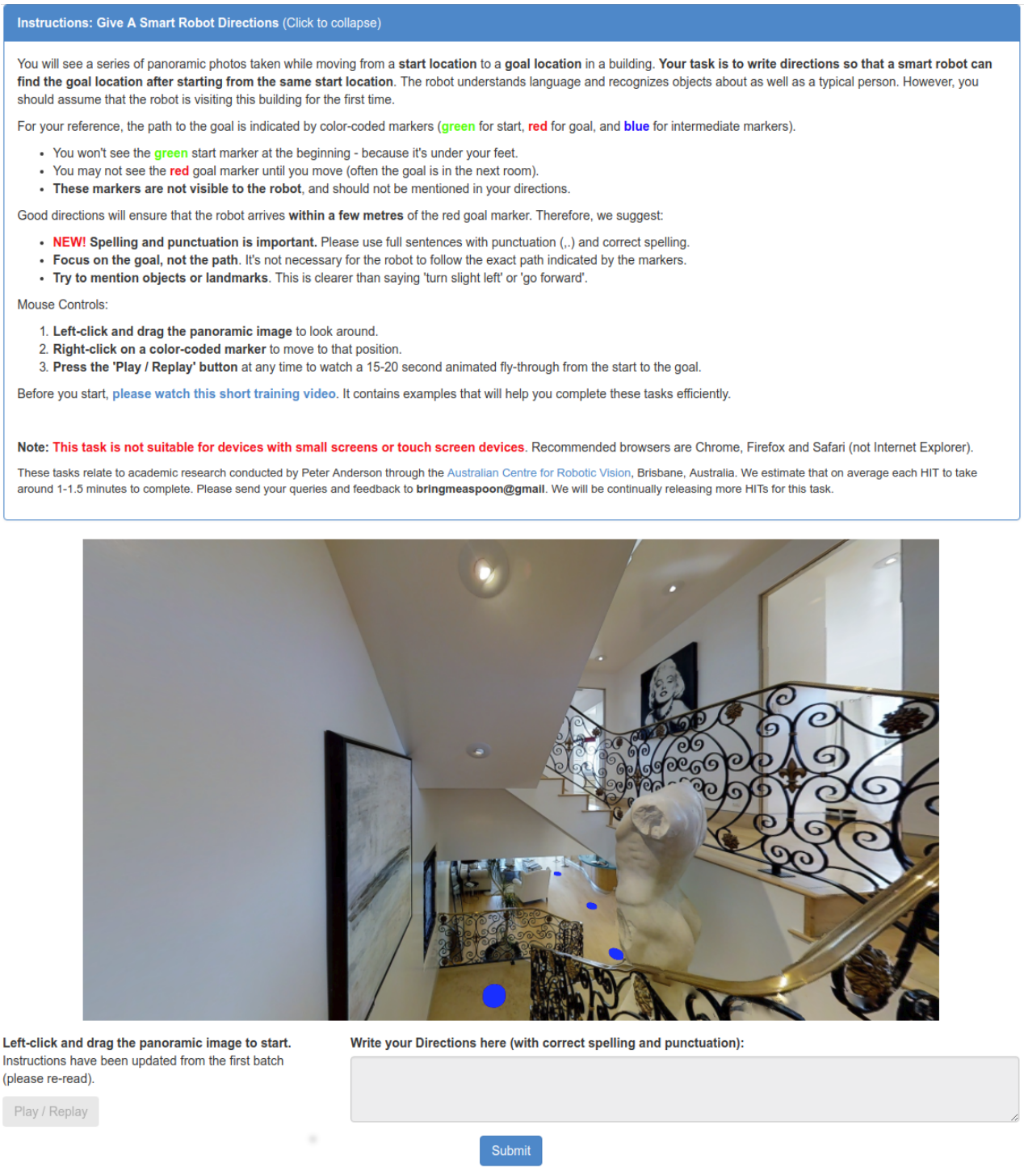

Figure 5: AMT data collection interface for the R2R navigation dataset. Here, blue markers can be seen indicating the trajectory to the goal location. However, in many cases the worker must first look around (pan and tilt) to find the markers. Clicking on a marker moves the camera to that location. Workers can also watch a 'fly-through' of the complete trajectory by clicking the Play / Replay button.