Following Formulaic Map Instructions in a Street Simulation Environment

Volkan Cirik*
Carnegie Mellon University
vcirik@cs.cmu.edu

Yuan Zhang
Google AI Language
zhangyua@google.com

Jason Baldridge
Google AI Language
jasonbaldridge@google.com

Abstract

We introduce a task and a learning environment for following navigational instructions in Google Street View. We sample $\sim$100k routes in 100 regions in 10 U.S cities. For each route, we obtain navigation instructions, build a connected graph of locations and the real-world images available at each location, and extract visual features. Evaluation of existing models shows that this setting offers a challenging benchmark for agents navigating with the help of language cues in real-world outdoor locations. They also highlight the need to have start-of-path orientation descriptions and end-of-path goal descriptions as well as route descriptions.

1 Introduction

The development of agents capable of providing and following navigational instructions has many practical applications and has the potential to drive significant advances in natural language understanding. Such applications necessitate research into linking language to the real world. Many new benchmarks linking language to the visual world have been created recently, including video captioning [1], image captioning [2, 3], referring expression recognition [4–6], visual question answering [7], and visual dialogue [8]. However, for these tasks, the perceptual input to the system is static i.e. the system's behavior does not change the perceived input. Recent studies propose more realistic scenarios where a system is asked to complete a task in a simulated environment where the perceptual input dynamically changes depending on the actions of the system. These environments either use a synthetic [9–12] or indoor environments [13, 14]. However, synthetic environments lower the complexity of visual scenes observed by the system. Realistic indoor environments also lack the chaotic nature of the visual world we observe on a daily basis. Outdoor learning environments [15–17], on the other hand, almost always guarantee that each scene consists of a unique combination of a high variety of objects. The transient nature of objects seen in an outdoor scene also poses several challenges for a system that is trained on an environment with snapshot images, but need to act in the real-world. However, we currently lack the large-scale data resources and environments to explore language for navigation in such real world settings that is commensurate with the complexity of the scenes and degrees of freedom involved.

This paper describes our preliminary efforts to build agents for simulated but messy real world environments with many degrees of freedom for movement, as part of our wider work on real world grounded language understanding [18]. To evaluate the capabilities of systems linking language into actions in an outdoor setting, a notoriously challenging real-world scenario, we introduce a novel task and an environment. In our setup, an agent is placed on a random starting point in a one-kilometer square region in a city and asked to navigate to a target point given an instruction sequence. The

*Research conducted during an internship at Google.

2nd Visually Grounded Interaction and Language Workshop (VIGIL-NIPS 2018), Montréal, Canada.

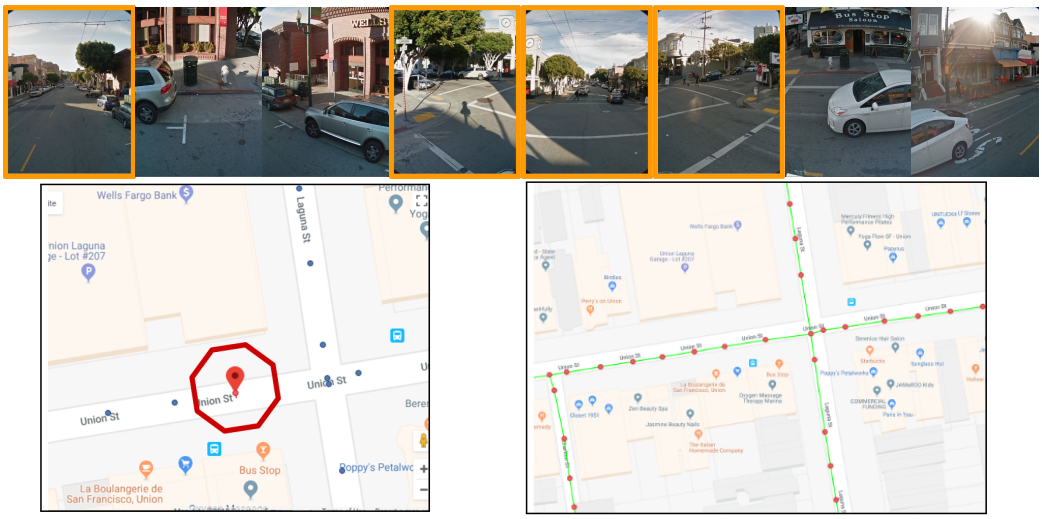


Figure 1: **Top**, The agent's visual input for a point. An orange frame indicates a navigable direction. Four directions are navigable in this example because the agent is at an intersection. **Bottom-Left**, The agent's real-world map location (the red octagon indicates the angles of image snapshots). **Bottom-right**, the connectivity of locations between navigable neighboring points.

actual coordinates and associated maps data are withheld from our agents, so they must rely on the visual environment and the language provided from the route description. At each point, the agent perceives eight images covering 360°perception at that physical point in the real-world and decides on moving to the next navigable point in its neighborhood. We evaluate baseline models from the literature [19] in this setting and find that their success rate is much lower than in a realistic indoor environment [13]. These results suggest outdoor instruction following poses several novel challenges to the vision, language, and robotics communities.

2 Street Simulator

We build an environment for agent navigation in a simulation of real-world outdoor settings. This requires real-world locations; for this, we use the publicly available Google Street View API[2] and Google Directions API.[3] The environment represents 1 km^2 regions of real-world locations. We sample a grid of locations spaced 20 meters apart using the Google Street View API and build a connectivity graph of undirected edges between them. At each location, we split the 360°panorama into eight images. Each image is connected to neighboring locations using the angle of the image and the distance between its location and the neighboring points. Using the graph, agents move from one location to another by choosing a navigable image. A sample map, the connectivity of locations, and the agent's point of view through images are in Figure 1.

Each navigation task consists of a start and end point, a path, and a sequence of navigation instructions. At each step, the agent has access to the instruction sequence and the current visual input, and it chooses one of the navigable images to move to. We created routes in 10 cities in the United States, with 10 manually selected regions per city (7 for training, 1 for development, and 2 for testing). We create tasks by sampling start and end locations that are at least 100 meters apart and obtain path instructions from the Google Directions API. Table 2 gives statistics for training and validation splits combined, a sample visualization of regions for Atlanta, Georgia,[4] and distributions of instruction and path lengths. Across all regions, the average number of steps is 38.6, with average length 770 meters, which is much greater than the Room-to-Room dataset's [13] average length of 10 meters (about 4-5 steps). The average number of tokens per path instruction is 27.6.

3 Experiments

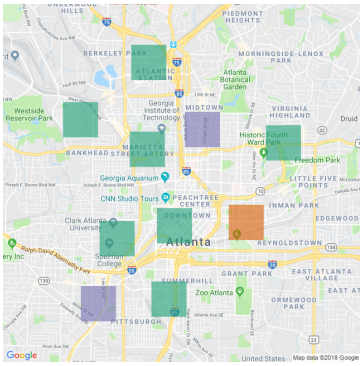
We experiment with a baseline model based on a sequence-to-sequence approach [13, 19]. Figure 3 shows an overview of the model. We encode the instruction of length L with a bi-directional LSTM [20, 21]. Specifically, given an instruction of length L, we embed each token of the instruction x_i to

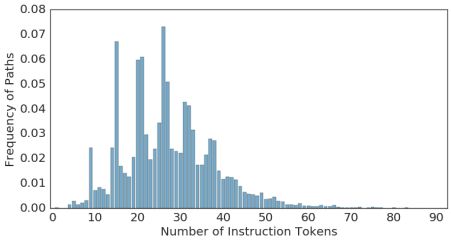
[2] `https://developers.google.com/maps/documentation/streetview/intro`

[3] `https://developers.google.com/maps/documentation/directions/start`

[4] Due to space limitations, we cannot provide statistics and visualizations for all cities.

City	Number of Points	Number of Paths	Vocabulary Size
Atlanta	4382	8000	408
Boston	6131	8000	659
Chicago	6566	8000	330
Los Angeles	6476	8000	314
New York	7366	7396	525
Philadelphia	8242	8000	367
Phoenix	5057	8000	220
San Diego	6931	8000	437
San Francisco	6964	8000	480
San Jose	4205	8000	318





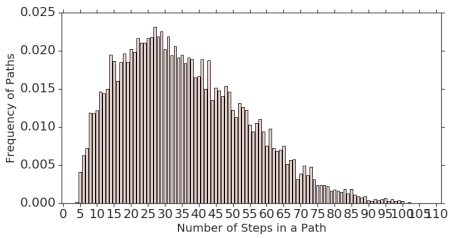


Figure 2: **Top-left**, The total number of points, the number of paths, and vocabulary size for a city in training and validation splits. **Top-right**, Visualization of 10 regions for Atlanta. Green, orange, and purple regions indicate training, validation, and test splits. **Bottom-left**, Distribution of instruction length for Atlanta regions. **Bottom-right**, distribution of path lenght in terms of the number of steps for Atlanta regions.

a vector and feed it to a two-layer LSTM. The final layer of the bi-LSTM forms the context vector $\hat{h}^{enc} = h_1^{enc}, h_2^{enc}, ..., h_L^{enc}$ for the action decoder that generate the sequences of actions.

We generate the sequences of actions with a decoder LSTM by processing the visual input, previously taken action, and dynamically attending to the context vector $\hat{h}^{enc}$ of the instruction. At each time step t, the decoder LSTM is fed with an embedding of the previously taken action and a visual sensory input to look at all snapshots. We calculate the visual-sensory input with an attention mechanism over the 8 snapshot images of different directions. The attention weights $a_{t,i}$ and attention probability $\alpha_{t,i}$ for each direction i is calculated with the decoder memory from the previous timestep h_{t-1}^{dec} as follows: $a_{t,i} = W_1 h_{t-1}^{dec} W_2 v_{t,i}$ and $\alpha_{t,i} = \exp(a_{t,i}) / \sum_i \exp(a_{t,i})$. The hidden state of the decoder LSTM h_t^{dec} is then used to attend the context representation of the instruction $\hat{h}^{enc}$ to induce the textual $\tilde{h}_t^{dec}$ for predicting an action . We calculate the probability of an action p_j to one of the 8 directions j using a bi-linear product $y_j = \left(W_3 \tilde{h}_t^{dec}\right)^T W_4 u_j$ and $p_j = \exp(y_j) / \sum_j \exp(y_j)$ where u_j is the concatenation of a convolution neural net, the vector representing the angle of the camera taking the snapshot image, and optical character recognition (OCR) output features for an image snapshot at direction j.

Implementation Details. We train a sequence-to-sequence model with stochastic gradient descent with a learning rate of 0.001 for 50 epochs for each city. The number of units for instruction embeddings, hidden state of encoder and decoder LSTM is set to 128.

Evaluation. We train agents for each city and measure four metrics on the validation split of all 10 cities. **Success rate** gives the percentage of times the agent navigates to within 40 meters of the target. **Oracle success rate** measures the percentage of agent paths that pass within 40 meters of the target. **Action accuracy** measures the performance of the agent for predicting the right action while navigating in 'teacher-forcing' regime [22, 13] where ground-truth actions are fed to the environment at each timestep. **Error distance** gives the average distance between the final location and the target.

Results. Figure 4 shows results for our experiments. For most of the source-target pairs, the agent's success rate is lower than 1%. Note that the same architecture achieves around 31.2% completion rate in an indoor environment when provided natural language descriptions. The low success rate

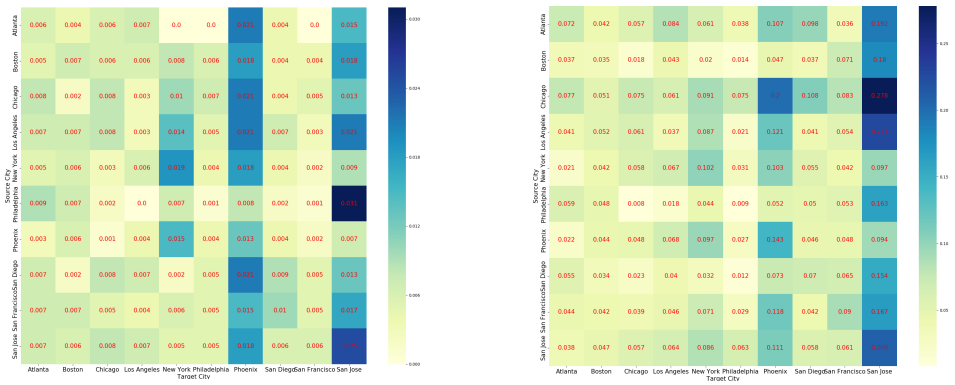


Figure 4: Success rate (**left**) and oracle success rate (**right**) for the sequence-to-sequence agent. The y-axis shows the source city for training, x-axis shows the target city for testing.

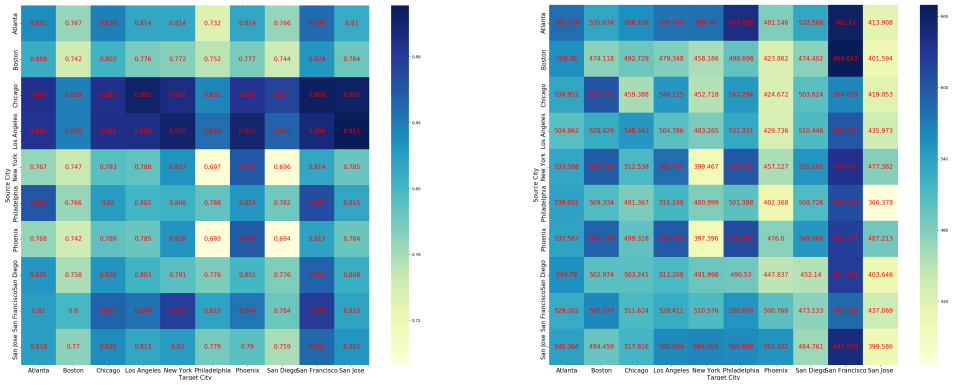


Figure 5: Action prediction accuracy (**left**) and average error distance (**right**).

of the baseline agent shows that outdoor scenes from the real-world pose a challenging scenario for navigation following agents. It also indicates that we need natural language descriptions of routes that are more directly connected to the visual environment. As with indoor environments, there is a large gap between the oracle success rate and the success rate for the baseline model: identifying where to stop remains an essential aspect of such navigation tasks.

Figure 5 shows action accuracy and the error distance. Since the average true path length is high for our routes, even over 90% action accuracy nonetheless results in high error distances. This results highlight the challenge of following a navigation instruction for a long path.

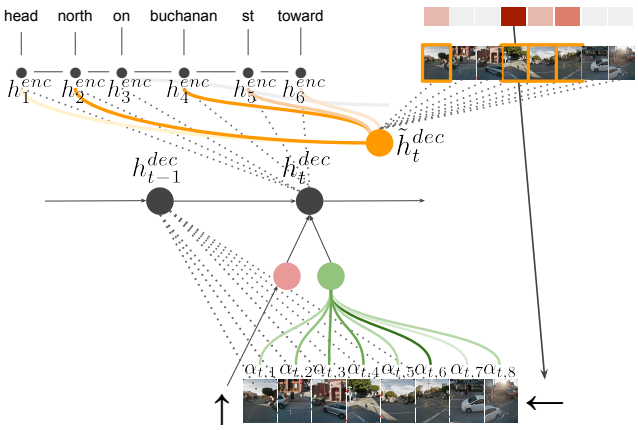


Figure 3: An overview of the model.

4 Conclusion

Our experiments show that outdoor scenes pose novel challenges for agents navigating in the real-world. Such an environment should benefit natural language processing, computer vision, and robotics community as a testbed for several navigation-based tasks, especially as more layers of annotation are added. In particular, the results from the preliminary experiments discussed in this paper indicate we need three distinct components to each path: (a) descriptions of initial heading to orient the agent and start moving in the right direction, (b) descriptions for the route and (c) a detailed description of the stopping point. These are all needed for successful completion of the Street View navigation task by human participants.

5 Acknowledgements

The authors would like to thank Eugene Ie and Slav Petrov for their valuable inputs.

References

[1] H. Yu and J. M. Siskind, "Grounded language learning from video described with sentences," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, pp. 53–63, 2013.

[2] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick, "Microsoft coco captions: Data collection and evaluation server," *arXiv preprint arXiv:1504.00325*, 2015.

[3] S. B. Oriol Vinyals, Alexander Toshev and D. Erhan, "Show and tell: A neural image caption generator," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3156–3164, 2015.

[4] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg, "Referit game: Referring to objects in photographs of natural scenes," in *EMNLP*, 2014.

[5] J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. Murphy, "Generation and comprehension of unambiguous object descriptions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11–20, 2016.

[6] H. De Vries, F. Strub, S. Chandar, O. Pietquin, H. Larochelle, and A. C. Courville, "Guesswhat?! visual object discovery through multi-modal dialogue.," in *CVPR*, vol. 1, p. 3, 2017.

[7] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, "Vqa: Visual question answering," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2425–2433, 2015.

[8] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra, "Visual Dialog," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[9] K. M. Hermann, F. Hill, S. Green, F. Wang, R. Faulkner, H. Soyer, D. Szepesvari, W. M. Czarnecki, M. Jaderberg, D. Teplyashin, *et al.*, "Grounded language learning in a simulated 3d world," *arXiv preprint arXiv:1706.06551*, 2017.

[10] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra, "Embodied question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[11] D. Gordon, A. Kembhavi, M. Rastegari, J. Redmon, D. Fox, and A. Farhadi, "Iqa: Visual question answering in interactive environments," *arXiv preprint arXiv:1712.03316*, vol. 1, 2017.

[12] M. Savva, A. X. Chang, A. Dosovitskiy, T. Funkhouser, and V. Koltun, "Minos: Multimodal indoor simulator for navigation in complex environments," *arXiv preprint arXiv:1712.03931*, 2017.

[13] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. v. d. Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[14] F. Xia, A. R. Zamir, Z.-Y. He, A. Sax, J. Malik, and S. Savarese, "Gibson env: real-world perception for embodied agents," in *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*, IEEE, 2018.

[15] P. Mirowski, M. K. Grimes, M. Malinowski, K. M. Hermann, K. Anderson, D. Teplyashin, K. Simonyan, K. Kavukcuoglu, A. Zisserman, and R. Hadsell, "Learning to navigate in cities without a map," *arXiv preprint arXiv:1804.00168*, 2018.

[16] S. Brahmbhatt and J. Hays, "Deepnav: Learning to navigate large cities," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 3087–3096, 2017.

[17] H. de Vries, K. Shuster, D. Batra, D. Parikh, J. Weston, and D. Kiela, "Talk the walk: Navigating new york city through grounded dialogue," *arXiv preprint arXiv:1807.03367*, 2018.

[18] J. Baldridge, T. Bedrax-Weiss, D. Luong, S. Narayanan, B. Pang, F. Pereira, R. Soricut, M. Tseng, and Y. Zhang, "Points, paths, and playscapes: Large-scale spatial language understanding tasks set in the real world," in *Proceedings of the First International Workshop on Spatial Language Understanding*, (New Orleans), pp. 46–52, Association for Computational Linguistics, June 2018.

[19] D. Fried, R. Hu, V. Cirik, A. Rohrbach, J. Andreas, L.-P. Morency, T. Berg-Kirkpatrick, K. Saenko, D. Klein, and T. Darrell, "Speaker-follower models for vision-and-language navigation.," *NIPS (to appear)*, 2018.

[20] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] A. M. Lamb, A. G. A. P. GOYAL, Y. Zhang, S. Zhang, A. C. Courville, and Y. Bengio, "Professor forcing: A new algorithm for training recurrent networks," in *Advances In Neural Information Processing Systems*, pp. 4601–4609, 2016.