Interpretable Agent for Language-guided 3D Indoor Navigation

Jiali Duan
EE546 Final Project
University of Southern California
jialidua@usc.edu

Abstract

Interpreting decision making is key to demonstrating agent's understanding of language semantics and its surroundings. In this project, we endow our agent with this ability by proposing a language synthesizer that is able to synthesize human instructions from its navigation trajectories and a contorller that is able to reliably execute instructions. The synthesizer module also serves as an alignment constraint between current observation and instruction, which reduces the possibility of misalignment in future time-steps. We show that our model is able to achieve state-of-the-art performance in challenging 3D indoor navigation environment. The demonstration video is available at https://sites.google.com/view/submission-2019.[1]

1. Introduction

Imagine a robot navigating across rooms following human instructions: "*Turn left and take a right at the table. Take a left at the painting and then take your first right. Wait next to the exercise equipment.*", the agent is expected to first execute the action "*turn left*" and then locates "*the table*" before "*taking a right*". However in practice, the agent might well turn right in the middle of the trajectory before a table is observed, in which case the follow-up navigation would definitely fail. Human on the other hand, has the ability to relate visual input with language semantics. In this example, human would locate visual landmarks such as table, painting, exercise equipment before making a decision (turn right, turn left and stop). In this paper, we endow our agent with similar reasoning ability by equiping our agent with a synthesizer module that implicitly aligns language semantics with visual observations.

Our insight is to exploit cycle-GAN framework [7], where $G_{A \rightarrow B}$ transforms input in domain A to domain B while $G_{B \rightarrow A}$ takes as input the generated output from $G_{A \rightarrow B}$ and transforms it back to the original domain A. By maximizing the joint likelihood of $G_{A \rightarrow B} * G_{B \rightarrow A}$, the two models can augment each other through collaboration. We use this strategy to interprete navigation decision into language instruction and transforms instruction into action at the same time. The two modules, which we dubbed synthesizer and controller respectively, help enforce alignment between language semantics and visual observations via maximum likelihood optimization.

2. Related Work

Language instruction following in unstructured environment has witnessed a surge in research interests after several release of benchmark dataset [1]. Its goal is to enable the robot to navigate across 3D environment given human language, either in the form of question or instructions. We use [1] as our dataset, which deals with human instructions with varying lengths and contexts. Another attribute of the environment used is that the layout of the room is diversified and complex, compared to previous environments where the agent only needs to explore either in one room or simplified rooms.

Our work can also be seen as a sequential decision making process, similar in spirit with reinforcement learning frameworks [4] where actions taken at previous timesteps exert influence for the future.

3. Method

3.1. Framework Components

Given a natural language instruction with L worlds, its representation is encoded as $X = (x_1, x_2 ... x_L)$ where x_l is the l-th word output by encoder LSTM. We follow the same strategy in [3] to use panoramic view as visual input, denoted as $V_t = (v_{t,1}, v_{t,2}, ..., v_{t,K})$, where K is the number of navigable directions and $v_{t,k}$ represents the image feature at direction k. At each time step t, a decoder LSTM takes grounded instruction $\hat{x_t}$, attended panoramic visual feature

[1] This work is for final project of EE546, project poster is available at https://davidsonic.github.io/summary/Poster_3d_indoor.pdf.

$\hat{v_t}$, previous action embedding a_{t-1} and previous hidden-state h_{t-1} as input, as Eqn 1.

$$h_t = LSTM([\hat{x_t}, \hat{v_t}, a_{t-1}, h_{t-1}]) \tag{1}$$

Textual grounding. The agent is expected to understand the context of instruction given current panoramic visual observations. We equip our agent with textual attention to help identify which part of language grounding is being utilized for indoor navigation. The attention weight over L words of the instruction is computed as:

$$z_{t,l}^{textual} = (W_x h_{t-1})^T x_l,\ \alpha_t = softmax(z_t^{textual}) \tag{2}$$

where W_x are parameters to be learnt. $z_{t,l}^{textual}$ denote the correlation between word l and previous hidden state h_{t-1} and α_t is the weight over textual features X at time t. Based on the attention distribution, the textual feature $\hat{x_t}$ is calculated as the weighted combination of textual representation $\hat{x_t} = \alpha_t^T X$.

Visual grounding. For each decision making, the agent needs to identify the most salient visual regions from current visual observations. We perform visual attention over image features from current views:

$$z_{t,k}^{visual} = (W_{v_1} h_{t-1})^T W_{v_2} v_{t,k},\ \beta_t = softmax(z_t^{visual}) \tag{3}$$

where W_{v_1} and W_{v_2} are parameters to be learnt. Similar to Eqn 2, the grounded visual feature $\hat{v_t}$ is the weighted combination of visual features $\hat{v_t} = \beta_t^T V$.

Action selection. Based on textual grounding, visual grounding above, action chosen at time step t is a bilinear dot product involving past history h_t and navigable action embedding at current step as Eqn 4.

$$y_t = (W_{o_1} h_t)^T W_{o_2} a_t,\ p_t = softmax(y_t) \tag{4}$$

The The agent then goes to next adjacent location by executing a_j with probability p_j.

3.2. Training

Our framework consists of a instruction synthesizer and action controller, responsible for interpreting human instructions and choosing action respectively. We train our model using a two-step procedure. First, we first pretrain our instruction synthesizer on the original dataset as follows:

$$L_{synthesizer} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{L} y_{i,t}^{syn} log P_{i,t,k} \tag{5}$$

where L is the maximum length of the instruction, N the batch size and k navigable directions at timestep t.

We then use the pretrained the synthesizer for data augmentation. Eventually, we train our action controller on the augmented data and then finetune on the original dataset.

$$L_{total} = \lambda L_{synthesizer} - (1-\lambda)\frac{1}{N}\sum_{i}^{N}\sum_{t}^{T} y_{i,t}^{ct} log P_{i,t,k} \tag{6}$$

where the total loss is a weighted sum of cross entropy from synthesizer as well as controller determined by λ.

4. Implementation Details

Image feature. Following [3], we use the pre-trained ResNet-152 on ImageNet to extract image appearance features. Therefore, the panoramic image feature at each timestep is 36 × 2048, where 2048 is the mean-pooled image feature with attention and 36=12 headings × 3 elevations with 30 degree intervals is the number of view angles. Here, the agent only needs to make high-leevl decisions as to which navigable direction to go next, instead of considering continuous action space, which is low-level visuomotor control.

The appearance feature is then concatenated with a 4-dim orientation feature [$sin\phi$,$cos\phi$,$sin\theta$,$cos\theta$], where ϕ and θ indicate the heading and elevation angles.

Network parameters. The length of each instruction is padded to 80. The word embedding used for navigation is 256 dimensional, initialized with GloVe embeddings [5]. A dropout layer with ratio 0.5 is appended after the embedding layer for regularization. The encoder/decoder LSTM has a hidden-state of 512 dimension. The textual grounding attention weight α_t is 80 dimensional while visual attention weight α_{vt} is 36 dimensional.

Submission to VLN challenge. For validating our proposed approach, we submit our result to Vison and Language Navigation challenge online test server. We achieved 55.67% (corresponds to Table 1) success rate on test-split.

We follow the guideline of submission rules, where all world states in the trajectories generated from beam-search were logged in the order they were traversed.

5. Experiments

R2R Dataset. We use the Room-to-Room (R2R) dataset [1] in our experiment, which contains 21,567 navigation instructions in total with an average length of 29 words. It's built upon the Matterport dataset [2], which contains 10,800 panoramic views from 194,400 RGB-D images of 90 building-scale scenes.

	Validation-Seen			Validation-Unseen			Test (unseen)		
Method	NE ↓	SR ↑	OSR ↑	NE ↓	SR ↑	OSR ↑	NE ↓	SR ↑	OSR ↑
Random	9.45	15.9	21.4	9.23	16.3	22.0	9.77	13.2	18.3
Student-forcing [1]	6.01	38.6	52.9	7.81	21.8	28.4	7.85	20.4	26.6
RPA [6]	5.56	42.9	52.6	7.65	24.6	31.8	7.53	25.3	32.5
Speaker-follower[3]	3.88	63.0	71.0	5.24	50.0	63.0	-	-	-
Speaker-follower(†)	3.08	70.1	78.3	4.83	54.6	65.2	4.87	53.5	96.0
Ours	3.26	67.58	74.93	4.91	53.26	64.96	-	-	-
Ours (†)	**2.88**	**71.79**	**80.80**	**4.76**	**54.79**	**67.65**	**4.57**	**55.67**	95.81

Table 1. Performance comparison of our method to previous work. NE is navigation error (in meters); lower is better. SR and OSR are success rate and oracle success rate (%) respectively (higher is better). † means with data augmentation.

Evaluation Protocol. We follow the same evaluation protocol as [1, 3] in our paper, where the (1) Navigation Error (NE) measures the shortest distance between the agent's final destination and the groundtruth destination; (2) Success Rate (SR) considers the percentage of navigations which end up with navigation error less than 3 meters; (3) Oracle Success Rate (OSR), the success rate if the agent can stop at the closes point to the goal along its trajectory.

5.1. Comparison with prior art

We compare our proposed approach with existing state-of-the-art as shown in Table 1. When trained with synthetic data, our method outperforms prior art by a margin of 2.1% in terms of success rate on the test set. We achieve 71.79% SR and 54.79% SR on validation-seen and validation-unseen respectively, compared to the best existing model which achieved 70.1% SR and 54.6% respectively. Without synthetic data, our model improves Speaker-Follower model [3] with 4.58% SR and 3.26% SR on validation-seen and validation-unseen respectively. Both results with or without data augmentation indicate that our proposed approach is more generalizable to unseen environments.

5.2. Qualitative Results

To further validate the proposed approach, we qualitatively show how our agent navigates the room given language instruction and visual observations at each timestep. Figure 1 shows the navigation process of our agent following human instruction. At each timestep, the agent is presented with panoramic observation space an the red arrow indicates the action chosen by our agent. At the end of the episode, the agent emits a "stop" to finish the navigation. More qualitative results are available at `http://mcl-lab.usc.edu:3000/trajectory.html.`

6. Conclusion

In this paper, we propose an interpretable navigation agent that aims to resolve the ambiguity between language semantics and visual observation in a photo-realistic 3D indoor environment. At each timestep, the controller executes the action which maximizes the similarity between the instruction given and the synthesized instruction. The synthesized instruction in turn, serves as an illustrator behind the action chosen. By involving the synthesizer and the controller in a loop, we are able to reduce the misalignment between visual inputs and instructions, which serves as an intuitive way for understanding machine's decision-making.

References

[1] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sunderhauf, I. Reid, S. Gould, and A. van den 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)*, volume 2, 2018. 1, 2, 3

[2] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Nießner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. 2

[3] 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. *arXiv preprint arXiv:1806.02724*, 2018. 1, 2, 3

[4] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016. 1

[5] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 2

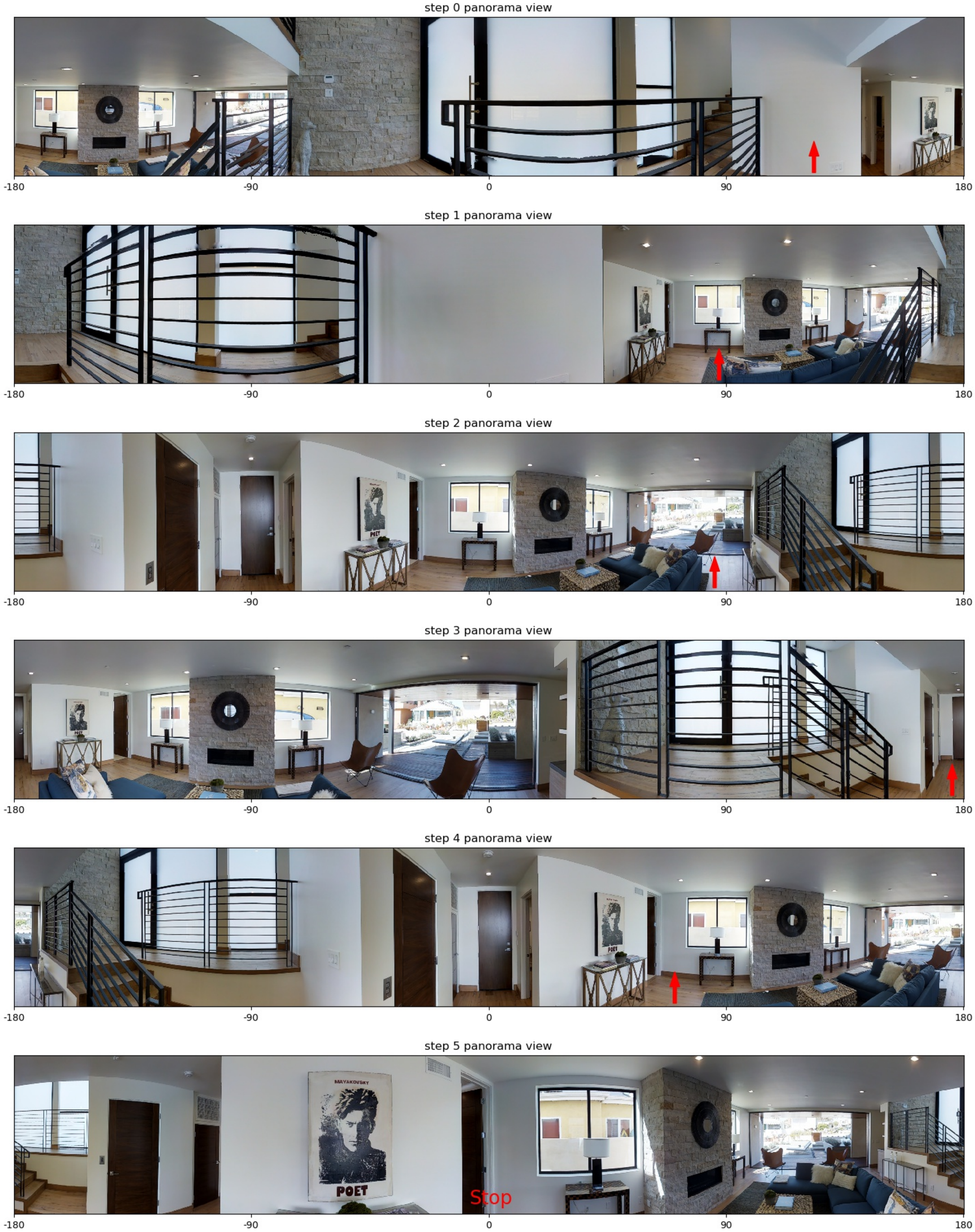
[6] X. Wang, W. Xiong, H. Wang, and W. Y. Wang. Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation. *arXiv preprint arXiv:1803.07729*, 2018. 3

[7] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017. 1

Instruction:

Walk down and turn right. Walk a bit, and turn right towards the door. Enter inside, and stop in front of a zebra striped rug.

rear: -180 degree *left*: -90 degree *front*: 0 degree *right*: +90 degree *rear*: +180 degree



Navigation steps of the panorama agent. The red arrow shows the direction chosen by the agent to go next.

Figure 1. Qualitative result of the model