# Generating Natural-Language Navigation Instructions from Panoramic Images

Erick Mendieta[1,2] Naoaki Okazaki [1,2] Hiroya Takamura[1,2]
[1]Tokyo Institute of Technology [2]AIST
{erick.mendieta,naoaki.okazaki} at nlp.c.titech.ac.jp
takamura.hiroya at aist.go.jp

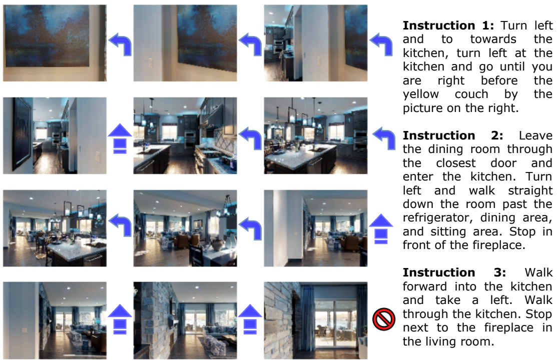
## 1 Introduction

The goal of the Vision-and-Language Navigation task (VLN) [1] is to train an agent to move through a lifelike environment using visual cues and a natural language instruction. The agent is required to understand the provided instruction and decide which action to take in order to get as close as possible to the target location. An example instruction can be found in Figure 1. The agent must be able to follow the commanded instruction even in previously unexplored environments.

Following navigation instructions is rather trivial for humans but hard for an agent [1, 4]. Humans tend to subjectively select reference points in the environment to describe a path. Therefore, we end up with a diverse set of instructions from the same sequence of images in the path. This can explain why it is difficult for an artificial intelligent agent to replicate humans in this task.

Previous approaches to this task [4] include a speaker-follower model. The speaker part of the model is a method for generating navigation instructions from panoramic images and actions in a path. The speaker model alone is useful because it can be used to interesting and important real-life applications such as autonomous driving navigation and virtual-reality navigation.

In this work, we focus on the navigation instruction generation problem. We extend the speaker model by taking into account the human behavior of selecting diverse reference points. In addition, we consider the fact that each view angle in a panoramic image corresponds to a time step on the simulator, representing actions Left, Right, Up, Down, Forward, and Stop.



**Instruction 1:** Turn left and to towards the kitchen, turn left at the kitchen and go until you are right before the yellow couch by the picture on the right.

**Instruction 2:** Leave the dining room through the closest door and enter the kitchen. Turn left and walk straight down the room past the refrigerator, dining area, and sitting area. Stop in front of the fireplace.

**Instruction 3:** Walk forward into the kitchen and take a left. Walk through the kitchen. Stop next to the fireplace in the living room.

**Figure 1:** An example path for the VLN task. The arrows beside the images represent actions in the environment (e.g., left, right, up, down, forward, stop).

Wang et al [10] noted this fact, and suggested that the VLN task has a sequential-decision making nature. The speaker model, however, uses an attention mechanism over all the viewpoints without considering the sequential nature of the task. To resolve this problem we create a panorama encoder which relies on a series of LSTM layers with attention to capture the sequential nature of panoramic images. The encoder is then used together with the Transformer encoder-decoder model [9] to generate navigation instructions.

## 2 Related Work

Four formal research papers have been presented so far to undertake this problem. The first work [1] introduced the VLN task, built the Room-2-Room (R2R) dataset, and presented a sequence-to-sequence baseline. Their focus was to improve the accuracy of task performance of the agent. This differs from our main goal, which is to produce natural-language navigation instructions from panoramic images. Wang et al [10] presented a recurrent policy network with look-ahead modules that look at the adjacent view angles to take advantage of the visual diversity of the panoramic images. However, similarly to Anderson et al. [1], they focused only on increasing the performance of the agent.

The third paper [4] proposed the Speaker-Follower approach, in which the speaker generates navigation instructions from panoramic images. The follower takes a decision by comparing the similarity scores between the generated instructions by the speaker and the golden instruction. In this paper, we use this work as a basis.

The last work [8] uses the attention mechanism in transformer [9] over the text to track the progress of the agent in the VLN task. They also use visual attention over the images to extract features, and give the result to a progress monitoring module. They do not generate instructions from panoramic images, which is the focus of our research.

# 3 Methodology

## 3.1 Panoramic action space

Similarly to [4], we define a panoramic action space. A path $P$ contains $n$ viewpoints $v_0, v_1, ..., v_n$. A viewpoint $v_i$ is formed by a discretized panoramic image, which has been divided into 36 view angles denoted by $k = 36$. Concretely, there are 3 elevations (top, middle, bottom) and 12 headings per elevation. In this work, we represent the view angles as elevation vectors $T = (t_0, t_1, ..., t_{11})$, $M = (m_0, m_1, ..., m_{11})$, $B = (b_0, b_1, ..., b_{11})$ for top, middle and bottom, respectively. With this notation, we represent a viewpoint $v_i = (T, M, B)$. A vertical segment $vs_i$ is defined by the top, middle and bottom view angles at position $i$; for example $vs_0 = (t_0, m_0, b_0)$. In total, we have 12 vertical segments. The action space contains 6 different possibilities: Left, Right, Up, Down, Forward, and Stop. We set the dimension $d$ of the hidden layer to 256.

## 3.2 Panoramic encoding model

The panoramic encoder consists of four left-to-right LSTM layers $L1, L2, L3, L4$. The first 3 layers $L1, L2, L3$ process the elevation vectors $T, M, B$ of the time steps independently of elevations, i.e., $(t_0, t_1, ..., t_{11})$ in $L1$, $(m_0, m_1, ..., m_{11})$ in $L2$, and $(b_0, b_1, ..., b_{11})$ in $L3$. The intention is to capture the sequential dependencies between each time step and its neighbors.

Then, the unrolled outputs of $L1, L2, L3$ are $A1, A2, A3$, respectively:

$$\begin{aligned} A1 &= (a1_0, a1_1, ..., a1_{11}), \\ A2 &= (a2_0, a2_1, ..., a2_{11}), \\ A3 &= (a3_0, a3_1, ..., a3_{11}). \end{aligned}$$

The outputs from the 3 layers are concatenated in order to form a vector $C = (A1; A2; A3)$. We then apply a Masked Multi-head attention mechanism (MMAH) with eight attention heads $h = 8$ as described in Transformer [9]. Here the query, key and value vectors $Q, K, V$ from the MMAH are all set to $C$. For the mask we have a vector IM $= (im_0, ...im_k)$, with $im_i = 1$ if we can access the next viewpoint from the $i$-th image, and $im_i = 0$ otherwise. Each $im$ value is determined by the output of the simulation environment. The output of MMAH is given as $(a1'_0; a1'_1; ...; a1'_{11}; a2'_0; a2'_1; ...; a2'_{11}; a3'_0; a3'_1; ...; a3'_{11})$.

For the last layer $L4$, we reshape the output of MMAH to form vertical segment headings $vs$ in the same way as described in Section 3.1. We obtain vector VS $= (vs_0, vs_1, ..., vs_{11})$ with $vs_i = (a1'_i, a2'_i, a3'_i)$, which is fed to $L4$ layer to obtain the vertical sequential relationships in the panoramic image with respect to its horizontal counterparts. A second Multi-head attention layer MMAV is used to attend the vertical relationships.

The next step uses a feed forward layer with size $(n * 12 * d, tsl * d)$ to reshape the output of MMAV to the required token sequence length $tsl = 50 + 1$. We add one token for the "<EOS>" token.

Finally, the output of the feed-forward layer is fed to the transformer model as described in the previous work [9]. We obtain the predicted tokens by performing the greedy decoding over the output.

# 4 Experiments

We first evaluate our model in terms of its ability to generate navigation instructions from panoramic images. We present an example from a randomly generated path of an unseen building. We then use BLEU

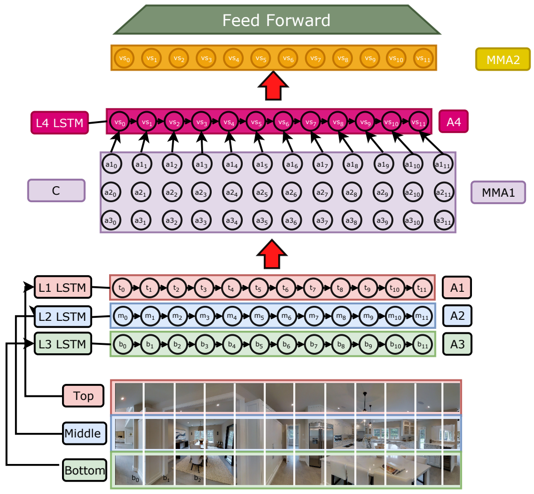


**Figure 2:** Panoramic encoding model.

| **Model** | **Val_seen** | **Val_unseen** |
|---|---:|---:|
| Speaker | 28.3 | **27.5** |
| Ours | **28.5** | 27.0 |

**Table 1:** BLEU score (%) of the baseline and the present work

score to compare our results with a baseline. We also evaluate the use of visual diversity of the panoramic image; we select a path in which multiple view angles can reach the next viewpoint. We compare generated instruction by the proposed method, the baseline, and humans (golden instructions).

**R2RDataset:** The Room-to-Room dataset [1] consists of 10,800 viewpoints constructed from 194,400 RGB-D images of 90 buildings from the matterport dataset [2]. In addition, selected 7,189 sample paths (5-7 viewpoints each) are associated with three navigation instructions (21,567 instructions in total) collected from Amazon Mechanical Turk (AMT). The average instruction length is 29 words. The vocabulary size is around 3.1k words (1.2k with 5 or more mentions). We use the dataset splits reported in [1]: training (14,025 instructions), validation seen (1,020), validation unseen (2,349) and test (4,173).

**Evaluation Metrics:** We report results with the standard BLEU score metric [6] in the validation seen and validation unseen splits. For the qualitative evaluation, we show a comparison of the generated sentences between the baseline, our model, and reference (golden).

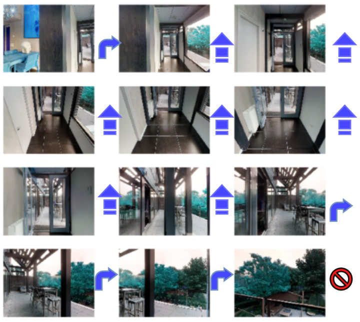
**Baseline:** As a baseline for our work, we selected the speaker model presented by Fried et al [4], whose purpose was to perform data augmentation and pragmatic inference in the VLN task.

**Implementation Details:** Following the previous research, we use the pretrained visual feature vectors provided by Anderson et al. [1]. These features originate from the final layer of a ResNet-152 [5] trained on ImageNet [3]. We use the negative log likelihood loss and Adam optimizer with default parameters. We use a batch size of 16, hidden size of 256. We train for 30,000 iterations. We use the greedy strategy for decoding.

## 5 Results and Analysis

We first report the comparison of the BLEU score. In terms of the BLEU score, our method performs better than the baseline in the validation seen split by a small margin. This small margin suggests that our method is comparable with the previous work. In the Validation unseen, however, our method performs slightly worse. We think that the reason for the small loss margin in the validation unseen is because we did not use pretrained GloVe embeddings [7] for the target words as the speaker model did. To further analyze the results, we perform a comparison between the good and bad examples predicted by the baseline and our work.

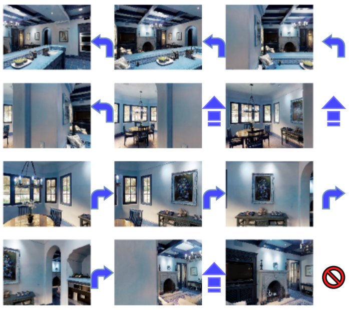
Similarly to the baseline, our method can generate instructions from paths in unknown buildings. The example with path id 1001358 comes from the data augmentation list provided by Fried et al. [4]. Figure 3 shows the instruction generated by our method. While these two sentences are similar to each other, ours tends to generate tokens that more closely refer to the objects seen in the picture. For example, in the same figure, our algorithm refers to the door as glass door, which is appropriate. Instead the previous research use the most generic term ”doorway”. We observe the same at the end of the sentence with the use of the phrase ”Wait on the porch”.



**Previous work**: walk down the hallway and turn right . walk down the hall and stop in the doorway .

**Our work**: turn around and go out the door . go out the glass door and go outside . wait on the porch .

**Figure 3:** Example 1001358 speaker augmentation split, Both algorithm are able to generate instructions for unseen case. However, ours better identify the objects by their names



**Golden:** Exit the kitchen heading towards the small dining nook and turn right. Continue forward and take the entrance ahead and to the right. Wait at the corner next to the end of both couches.

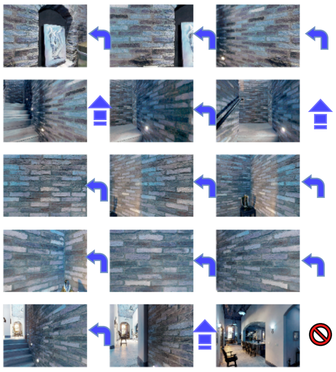
**Previous work:** turn around and walk through the kitchen . turn right and walk past the dining table . stop in front of the dining table .

**Our work:** walk into the kitchen and turn right . walk past the table and into the living room . stop in front of the fireplace .

**Figure 4:** Example 2068-0 validation unseen. Our method predicts "fireplace" found in the images, although it is not mentioned in the golden navigation instruction.

Our method also seems to recognize objects that are not present in the golden instruction but are shown in the image. Figure 4 provides such an example, in which the word "fireplace" appears in our prediction, but not in either the golden instruction nor the generated instruction generated by the previous work.

More confusing examples are the ones that contains a repetition of actions. For example, when there is a staircase with multiple stair flights to climb, the algorithm in the previous work tends to repeat the same instruction over and over again. The same thing happens when the environment contains multiple doors in the same place. Our method, however, benefits from the sequential nature of the task and the use of panoramic information to better represent the scenario. See Figure 5.



**Our work**: walk up the stairs . walk past the pillars on the left . wait at the edge of the rug .

**Previous work**: go up the stairs and go up the stairs . go up the stairs and stop at the top of the stairs .

**Golden instruction:** Ascend stairs, then walk toward the kitchen area, and wait.

**Figure 5:** Example 5412-0 Validation unseen. We show the most difficult examples for which our method has a slightly better advantage by making use of the sequential nature of the panoramic images.

## 6 Conclusion

We presented a method for generating navigation instructions from panoramic images that takes advantage of the sequential nature of the VLN task.

**Acknowledgment** This work is supported by RWBC-OIL.

## References

[1] Peter Anderson et al. "Vision-and-Language Navigation: Interpreting visually-grounded navigation instructions in real environments". In: *CVPR*. 2018.

[2] Angel Chang et al. "Matterport3D: Learning from RGB-D Data in Indoor Environments". In: *3DV*. 2017.

[3] Jia Deng et al. "ImageNet: A Large-Scale Hierarchical Image Database". In: *CVPR*. 2009.

[4] Fried et al. "Speaker-Follower Models for Vision-and-Language Navigation". In: *NeurIPS*. 2018.

[5] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *CVPR*. 2015.

[6] Kishore Papineni et al. "BLEU: a Method for Automatic Evaluation of Machine Translation". In: *ACL*. 2002.

[7] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation". In: *EMNLP*. 2014.

[8] anonymous - under review. "Self-Monitoring Navigation Agent via Auxiliary Progress Estimation". In: *ICLR*. 2019.

[9] Ashish Vaswani et al. "Attention Is All You Need". In: *NeurIPS*. 2017.

[10] Xin Wang et al. "Look Before You Leap: Bridging Model-Free and Model-Based Reinforcement Learning for Planned-Ahead Vision-and-Language Navigation". In: *ECCV*. 2018.