Visual Landmark Selection for Generating Grounded and Interpretable Navigation Instructions

Sanyam Agarwal[1] Devi Parikh[1,2] Dhruv Batra[1,2]
Peter Anderson[1] Stefan Lee[1]
[1]Georgia Institute of Technology, [2]Facebook AI Research
[1]{sagarwal344, parikh, dbatra, peter.anderson, steflee}@gatech.edu

Abstract

Instruction following for vision-and-language navigation (VLN) has prompted significant research efforts developing more powerful "follower" models since its inception in [1]; however, the inverse task of ***generating visually grounded instructions given a trajectory*** *– or learning a "speaker" model – has been largely unexamined. This task is itself a challenging visually-grounded language generation problem akin to video or image captioning. Unlike these tasks however, instruction generation has a straightforward notion of correctness – can a follower arrive at the correct location based on generated instructions? Further, improved speaker models can be leveraged to strengthen follower models via data augmentation or back-translation.*

In this abstract we present a work-in-progress "speaker" model that generates navigation instructions in two stages, by first selecting a series of discrete visual landmarks along a trajectory using hard attention, and then second generating language instructions conditioned on these landmarks. This two-stage approach improves over prior work, while also permitting greater interpretability. We hope to extend this to a reinforcement learning setting where landmark selection is optimized to maximize a follower's performance without disrupting the model's language fluency.

1. Introduction

Lost trying to find Jane's office, you call a friend familiar with the building who says "Take a left from the lobby and go down the hallway with a large painting at the end. Jane's office is on the left opposite the break room." To generate this instruction, the "speaker" composed a series of actions ("take a left", "go down the hallway") grounded in visual content the "follower" would observe (e.g. "lobby", "large painting"). The key question in this work is how to develop models that mirror this capability in realistic environments.

This task – which we refer to as *trajectory-grounded instruction generation* – is a challenging visually-grounded language generation problem. Concretely, given a trajectory through an environment consisting of both the visual perceptions and actions associated with each time step, the task is to produce a natural language instruction that succinctly conveys the information necessary to reproduce the trajectory. In addition to its research merit as a step towards improved human-robot collaboration, this task is relevant to a number of immediate practical applications such as navigation apps that provide audible directions and do not require users to look at an on-screen map.

Unlike related tasks such as image and video captioning, trajectory-grounded instruction generation has a pragmatic evaluation metric – can a follower arrive at the correct location given the instructions? Further, this task also differs in the fact that the vast majority of visual content observed along a trajectory is irrelevant to the generated instructions. Consider our initial example; only a few visual concepts were referenced ("lobby", "painting", "break room"), based both on their visual saliency (i.e. likelihood of being noticed) and their utility as efficient landmarks for navigation.

In an effort to model this problem structure, we decompose the task of generating instructions into two stages – first selecting salient landmarks in the trajectory through a hard-attention mechanism, and then using these landmarks to generate the instruction via a sequence-to-sequence model with soft attention. Notably we also observe that given the trajectory including actions, landmark selection can be performed independently for each time step. Decoupling these stages permits greater intepretability in the model's visual grounding while simultaneously improving over a standard sequence-to-sequence approach [5] in our experiments. This structure also allows for direct intervention experiments to change the focus of instructions and examine the extent of the visual grounding.

Looking forward, our landmark selection model provides a compact decision space for a reinforcement learning agent to operate on. With a fixed language generation model, such an agent could optimize landmark selection to

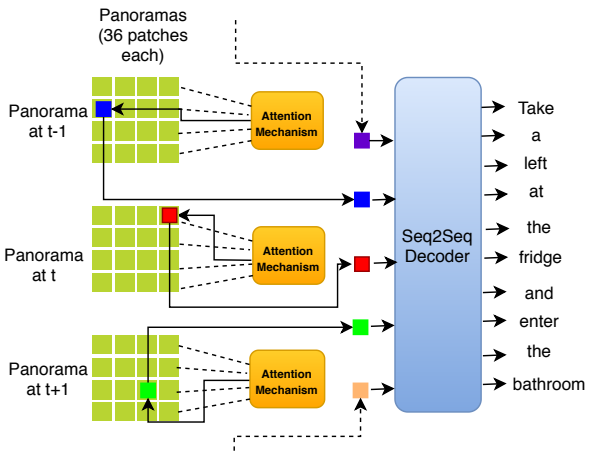

Figure 1. Proposed model illustrating visual landmark selection with hard attention (left) and landmark to instruction decoding using Seq2Seq with soft attention (right).

improve a follower's success without having to grapple with the significantly larger vocabulary space and the problem of radical divergence from natural language (i.e. linguistic drift). We leave this as future work.

2. Related work

Vision-and-Language Navigation. Significant efforts have been devoted to the problem of following natural language navigation instructions in perceptually realistic virtual environments [1, 5, 18, 11, 10, 8, 15]. Much of the recent work in this area has been tied to the Room-2-Room (R2R) instruction dataset presented in [1]. This dataset consists of ∼7200 trajectories through Matterport3D [2] virtual environments each with three human-annotated instructions for following the trajectory. While originally collected for the instruction following task, we utilize this dataset to learn to generate trajectory-grounded instructions.

Visually-grounded Instruction Generation. Compared to the problem of following instructions, comparatively less recent attention has been devoted to the inverse problem of visually-grounded instruction generation [5, 3]. However, several works have demonstrated that instruction generation (speaker) models can be leveraged to significantly improve follower models via data augmentation or back-translation [5, 15], which further motivates our work.

3. Approach

Our speaker model consists of a hard-attention based visual landmark encoder that selects important landmarks and a sequence-to-sequence instruction decoder that transforms these landmarks into grounded instructions (refer Figure 1).

3.1. Encoding Trajectories

Notation. A navigation trajectory x is an ordered sequence of panoramic views $(p_1, p_2, p_3, ..., p_m)$ where m is the number of total steps in x. Note we also assume relative information between the panoramas such as direction and distance, which is reasonable for an agent actually traveling this trajectory. Further, trajectory x is paired with a natural language instructions $y = (w_1, w_2, \ldots, w_T)$. While there are three instructions per trajectory, we ignore this for notational clarity but do consider all instructions during training.

Visual Representation. Following prior work [5], we divide the panoramic sphere into 36 patches which covering three level of elevations each with 12 equally spaced patches covering 360 degrees horizontally. As in [1], we pre-compute a 2048-dimensional visual feature for each patch from the final convolutional layer of a frozen ResNet [6] pretrained on ImageNet [14]. As such, each panorama p_i is described by a 36×2048 matrix V_i where each row $V_i^{(j)}$ is a visual feature vector for one of the 36 patches in the panoramic view.

Directional Encodings. There is a great deal of directional information in a trajectory that our model also needs to reason about – such as heading changes and relative angles – in order to generate instructions like "turn left" or "opposite the break room". For each patch j in panorama p_i, we have the corresponding direction $d_i^{(j)}=[h_i^{(j)}, e_i^{(j)}]$ consisting of the patch's heading and elevation angles $h_i^{(j)}$ and $e_i^{(j)}$. Further, we assume access to the incoming and outgoing directions to neighbouring panoramas $d_i^{(in)}=[h_i^{(\text{in})}, e_i^{(\text{in})}]$ and $d_i^{(out)}=[h_i^{(\text{out})}, e_i^{(\text{out})}]$. To represent angular differences, we introduce an encoding function $f(d^{(j)}, d^{(k)})$ defined as

$$f(d^{(j)}, d^{(k)}) = \Big[sin(h^{(j)} - h^{(k)}), cos(h^{(j)} - h^{(k)}), sin(e^{(j)} - e^{(k)}), cos(e^{(j)} - e^{(k)})\Big] \quad (1)$$

We extend the visual feature matrix V_i by concatenating relative direction encodings with the incoming and outgoing directions for each patch as well as a fixed encoding between the incoming and outgoing directions. We denote this augmented matrix as D_i and can write its j^{th} row as

$$D_i^{(j)} = \Big[V_i^{(j)} : f(d^{(j)}, d^{(\text{out})}) : f(d^{(j)}, d^{(\text{in})}) : f(d^{(\text{in})}, d^{(\text{out})})\Big] \quad (2)$$

where we use : to denote vector concatenation. In practice, we find replicating the directional encodings useful and repeat each 32 times when forming $D_i^{(j)}$ above.

3.2. Visual Landmark Selection

Now that we have encoded our trajectory into a series of matrices $(D_1, D_2, ..., D_m)$ containing direction-augmented visual features, we would like to learn to identify important landmarks to guide instruction generation. To select landmarks, we perform hard self-attention independently on each panorama – outputting one landmark (patch) l_i from

	val-seen							val-unseen						
Method	Bleu-1	Bleu-4	CIDEr	Meteor	Rouge	SPICE	Follower	Bleu-1	Bleu-4	CIDEr	Meteor	Rouge	SPICE	Follower
SF Speaker [5]	0.537	0.155	0.121	0.233	0.350	0.203	0.491	0.522	0.142	0.114	0.228	0.346	0.188	0.273
Softmax	**0.549**	0.156	0.132	0.233	0.354	**0.214**	**0.501**	**0.548**	0.157	0.129	**0.231**	**0.357**	**0.199**	0.270
Gumbel Softmax	0.541	**0.157**	0.134	**0.234**	**0.356**	0.213	0.487	0.529	0.150	0.125	0.229	0.353	0.191	**0.281**
2-Phase	**0.549**	**0.157**	**0.137**	0.228	0.352	**0.214**	0.492	**0.548**	**0.159**	**0.132**	**0.231**	**0.357**	0.197	0.272

Table 1. Instruction generation performance on the R2R validation sets. For "SF speaker" row we use the "speaker" model released by [5]. For all other rows we report the average of 3 runs in each column. We find that our model improves over prior work across both splits, and 2-Phase training produces a hard attention model with comparable performance to softmax attention.

each panorama p_i based on features D_i.

To give the self-attention mechanism context of the next trajectory step, we define a matrix C_i as the concatenation of D_i and a tiling of $D_i^{(\text{out})}$ (the feature for the patch most aligned with the outgoing direction), such that each row of C_i can be written $C_i^{(j)} = [D_i^{(j)} : D_i^{(\text{out})}]$. We pass C_i to a series of three transformer-style self-attention blocks which we do not describe in detail here for sake of space. In essence, these modules learn to iteratively refine query vectors through attention-based interactions – please see [16] for full details. After the third round of attention, we take the average query vector q_{avg} and compute a final attention vector over embeddings of the direction-augmented visual features given by:

$$a = \text{softmax}\left((q_{avg}^T \circ K)/\sqrt{d_q}\right) \quad (3)$$

$$K = D_i^T W_{fc} + b_{fc} \quad (4)$$

where d_q it length of q_{avg} and $\circ$ is a broadcasted dot product. We can then select the final output landmark as

$$l_i = K^T \cdot \mathbf{1}[\text{argmax } a] \quad (5)$$

where $\mathbf{1}[x]$ is a one-hot vector with 1 at the x^{th} position. As the argmax operation is not differentiable, we train using the Gumbel softmax straight-through estimator from [7, 12].

We perform this attention for each panorama yielding a sequence of landmarks. Notably, each landmark is just a linear embedding of the direction-augmented feature representation of a panorama patch. This means that while significant mixing occurs during the self-attention process, the final output to the instruction generation model is simply one of the initial observations.

3.3. Instruction Decoding

To generate the final instruction from a sequence of landmarks, we use a sequence-to-sequence model with soft attention similarly to [5]. Contextual information between the landmarks is captured through a bidirectional LSTM encoder that takes the landmark sequence and produces hidden states $h_1, h_2, ..., h_m$. An LSTM decoder then generates the instruction by attending over the encoder hidden states at each time step before selecting the output word. The decoder then updates it state and repeats this process until the end of sentence token is generated.

4. Experiments

Dataset. We evaluate in the Room-to-Room (R2R) instruction dataset presented in [1]. The dataset consists of ∼21,500 natural language instructions corresponding to $\sim 7,200$ trajectories through the virtual 3D environments of Matteport3D [2]. The dataset is split into training, val-seen, val-unseen, and test sets. We evaluate on val-seen and val-unseen which correspond to new trajectories in either seen or unseen environments.

Evaluation Metrics To evaluate generated instructions we examine captioning metrics as well as follower success.

- **Fluency**: To evaluate agreement with the 3 human-written instructions associated with each trajectory, we use standard image caption evaluation metrics such as Bleu [13], Meteor [4], Cider [17], Rouge [9], Spice [9].
- **Follower**: We also report the success rate of a follower given the generated instructions. In place of a human follower, we use a model from prior work [5]. For a fair comparison of speaker performance, we skip the "Speaker-Driven Data Augmentation" and "Pragmatic Inference" training steps that co-train the speaker and follower model in their approach.

Higher scores on the follower metric would suggest that the instruction is more understandable and useful for a human trying to follow that trajectory which is what we care about.

Training. We train our model to minimize cross-entropy loss of ground-truth human instructions. In addition to directly training our hard-attention model **(Gumbel Softmax)**, we also examine a soft-attention model, i.e $l_i = K^T \cdot a$, **(Softmax)** and a hard-attention model initialized from a soft-attention model after $20k$ iterations **(2-Phase)**.

4.1. Results

Table 1 shows results for the R2R validation sets for our approach and a baseline sequence-to-sequence model from [5]. We find that our model with soft-attention outperforms the baseline on almost all metrics across both splits

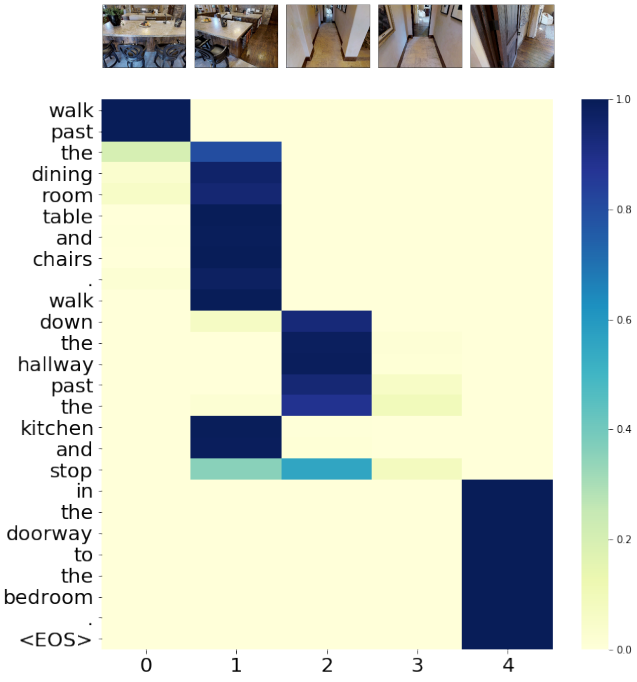

Figure 2. Soft attention grounding between generated instruction words and the selected visual landmarks along the input trajectory.

– indicating that the independent self-attention and directional encoding strategy we employ is effective at this task generally. Further, we find that the 2-Phase training approach improves significantly over training with Gumbel softmax from scratch, producing a hard attention model with comparable performance to our soft-attention variant. This improves interpretability and opens up significant opportunities for future work using reinforcement learning approaches to optimize landmark selection while maintaining the fluency of the generated instructions.

Visual Grounding. We show an example of visual grounding for the instructions generated by our model in Figure 2. The 5 images at top are visual landmarks selected by our hard attention model from 36 viewpoints at each step in the input trajectory. The heatmap illustrates the soft attention weights in the instruction decoder, illustrating correct visual grounding for phrases such as "dining room table and chairs", "hallway", and "doorway to the bedroom".

References

[1] 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. In *CVPR*, 2018. 1, 2, 3

[2] 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. In *3DV*, 2017. 2, 3

[3] Andrea F Daniele, Mohit Bansal, and Matthew R Walter. Navigational instruction generation as inverse reinforcement learning with neural machine translation. In *HRI*. IEEE, 2017. 2

[4] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*, 2014. 3

[5] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. In *NeurIPS*, 2018. 1, 2, 3

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2

[7] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv*, 2016. 3

[8] Liyiming Ke, Xiujun Li, Yonatan Bisk, Ari Holtzman, Zhe Gan, Jingjing Liu, Jianfeng Gao, Yejin Choi, and Siddhartha Srinivasa. Tactical rewind: Self-correction via backtracking in vision-and-language navigation. In *CVPR*, 2019. 2

[9] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004. 3

[10] Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. Self-monitoring navigation agent via auxiliary progress estimation. In *ICLR*, 2019. 2

[11] Chih-Yao Ma, Zuxuan Wu, Ghassan AlRegib, Caiming Xiong, and Zsolt Kira. The regretful agent: Heuristic-aided navigation through progress estimation. In *CVPR*, 2019. 2

[12] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv*, 2016. 3

[13] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 3

[14] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 2

[15] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. In *NAACL*, 2019. 2

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 3

[17] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015. 3

[18] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. *arXiv*, 2018. 2