Technical Report for REVERIE and SOON Object-oriented Vision-and-Language Navigation Challenges (ICCV 2021)

Shizhe Chen†, Pierre-Louis Guhur†, Makarand Tapaswi‡, Cordelia Schmid† and Ivan Laptev†
†Inria, École normale supérieure, CNRS, PSL Research University ‡IIIT Hyderabad
shizhe.chen@inria.fr

Abstract

In this report, we present our approach for the ICCV 2021 REVERIE and SOON challenges. The two challenges tackle the object-oriented vision-and-language navigation (VLN) task. Given a natural language sentence about a desired object, the goal of this task is to explore an environment, arrive at a target viewpoint, and localize the target object. We present a transformer-based model with graphical map (GMAP) for the task. It disentangles the navigation policy and stop policy. The navigation policy constructs graphical maps on-the-fly to improve exploration in unseen environment, and the stop policy utilizes pretrained transformers and auxiliary tasks to improve generalization. This novel model achieves top performance in the REVERIE and SOON challenges, with more than 60% relative improvements over existing methods.

1. Introduction

The goal of vision-and-language navigation (VLN) task [1] is to learn navigational agents that can follow human natural language instruction. It requires the agent to perceive the visual world, understand natural language and make actions based on the multimodal inputs. Most of recent works [6, 8, 10, 11, 14, 15, 19–21] are focusing on VLN with fine-grained instructions (*e.g.* R2R [3], RxR [12], Touchdown [5] and ALFRED [17] datasets), which provide step-by-step navigation guidance, for example *"Go out the door in front of you. Once in the hallway turn left. Walk forward and then turn left into the sitting area. Stop in front of the fireplace"*. Such fine-grained instructions simplify the navigation task, so that the agent only needs to faithfully follow the path without exploring the environment on its own. However, this type of instructions is cumbersome for real life applications. A simpler way to interact with agents are high-level instructions such as *"Go to the lounge room and pick up the top picture above the lamp"*, which mainly describe the target location and object.

The ICCV 2021 REVERIE [16] and SOON [22] challenges tackle the latter type of instructions, which we dub object-oriented VLN task. The task aims at navigating to a target location and localizing a desired object in the scene given a high-level instruction. The navigation performance alone can be measured by Success Rate (SR, the percentage of cases where the agent arrives at the target location), Oracle Success Rate (OSR, the percentage of cases where the agent has passed by the target location) and SPL (SR weighted by Path Length to penalize long exploration paths). The final object localization performance is evaluated by Remote Grounding Success (RGS, the percentage of cases where the agent arrives at the target location and correctly localizes the object) and RGSPL (RGS weighted by Path Length). As object localization accuracy and navigation efficiency are both important for object-oriented VLN, the primary metric in the challenges is RGSPL.

The object-oriented VLN task poses several new challenges compared to VLN with fine-grained instructions [3] or object-goal navigation [4]. *Firstly*, since there is no step-by-step navigation guidance, *the agent needs to learn efficient exploration strategy in unseen environments* to find the target. It should maintain a long-term memory of explored scenes in order to more efficiently navigate to unvisited areas. Furthermore, commonsense knowledge is necessary for the agent to make smarter navigation decisions, for example, the common room arrangement. *Secondly, the task requires cross-modal reasoning abilities to decide where to stop and which object to localize*. The instruction not only identifies the room/object types, but also contains more fine-grained attributes and relationships to specify the exact instance. Moreover, the object grounding is performed in panoramic view which contains more objects and complex spatial relationships.

To address the above challenges, we propose a transformer-based model with a graphical map (GMAP) for the object-oriented VLN task. We disentangle the overall decision making into a stop policy and a navigation policy. The stop policy decides whether to stop in a viewpoint and localize the object. The navigation policy aims to explore the unseen viewpoints based on a graphical map built on-the-fly, which is able to combine planning and learning for more efficient exploration. Both policies utilize transformer architectures and benefit from vision-and-language pretraining. Our proposed GMAP model outperforms prior methods on the object-oriented VLN task by a large margin and won the ICCV 2021 REVERIE and SOON challenges.

2. Method

Our proposed model contains two policy networks: stop policy and navigation policy. At each step, the stop policy predicts a probability to stop at the current viewpoint. If it decides so, it selects an object in the viewpoint; otherwise the navigation policy is applied. The navigation policy builds a graphical map on-the-fly, and predicts the next viewpoint to explore. It uses the shortest path planned from the graphical map to navigate to this viewpoint. If it exhausts the maximum number of action steps, the agent is forced to stop, and it uses the map to navigate to a viewpoint with maximum stop probability as its final prediction. We present the two policies in detail below.

2.1. Stop Policy Network

The stop policy takes as input an instruction and the panoramic image of a given viewpoint. It outputs whether to stop at the viewpoint or not. If it stops, then the policy predicts which object is the target one.

Model Architecture. The stop policy network is built upon LXMERT [18], a pretrained vision and language transformer. It first encodes the instruction and the panoramic image independently, and then encodes the cross-modal relationships of the text and image. Specifically, the instruction is embedded through the look-up table and fed into a multi-layer transformer to obtain textual representations. For the panoramic image, we extract global visual features of 36 views in the panorama via a vision transformer [7] as well as local visual features of objects in the panorama given annotated or detected object bounding boxes [2]. The visual features are added with position embeddings, and fed into another transformer to obtain visual representations. A special `stop` token is concatenated with visual tokens to denote the stop action. As geometric location of a viewpoint is beneficial to decide whether to stop, we add location embeddings of the viewpoint to the above visual representations. Next, a multi-layer transformer with cross-modal attentions is used to jointly encode the textual and visual representations. Finally, a stop prediction head takes the output embeddings of the `stop` token and all navigable views as input and predicts whether to stop or navigate to another direction. A object grounding head takes the output embeddings of all objects to predict the target object. Both prediction heads are implemented as multi-layer perceptrons.

Training. The stop policy network is trained by classification losses for the stop action prediction task and the object grounding task. To alleviate overfitting, we also train it with several auxiliary tasks, including masked language modeling and masked region modeling.

Ensembling with an object grounding module. In order to further improve the object grounding performance, we train another object grounding (OG) module based on LXMERT and adopt a late pooling prediction strategy as in [13]. The module is firstly trained on several proxy tasks including masked language modeling, masked region modeling and negative instruction matching [9]. Then, the OG module is finetuned on the object grounding task.

2.2. Navigation Policy

We construct a graphical map on-the-fly during the navigation. Each node in the graph denotes a viewpoint and the edge denotes the connectivity of viewpoints. The navigation policy network is fed with all unvisited viewpoints in the graphical map and the instruction. It predicts the most promising viewpoint to reach the target.

Model Architecture. We reuse the textual and view visual representation from the stop policy network for the instruction and node embeddings. The location of each viewpoint is also added to the node embedding. We utilize the same cross-modal transformer architecture as in the stop policy to encode the text and the graph. An action prediction head is used to predict the next viewpoint to go.

Training. We combine teacher forcing and student forcing training algorithms to train the navigation policy. In the student forcing training, the best next viewpoint is selected by the summarized distance to the current viewpoint and the target viewpoint, which can avoid traveling to a too distance viewpoint from the current position.

3. Experiments

3.1. Datasets

REVERIE. The REVERIE dataset provides object bounding boxes (but not their labels) for each viewpoint. The task, hence, requires to select the target from all annotated objects in the viewpoint. The dataset is split into training, val seen, val unseen, and test sets. The training set contains 60 scenes and 10,466 instructions; the val seen set consists of 46 scenes and 1,423 instructions; the val unseen set includes 10 scenes and 3,521 instructions; and for test set, there are 16 scenes and 6,292 instructions. The path length ranges from 4 to 7 viewpoints and there are 21 tokens on average in the instruction.

SOON. The SOON dataset does not contain annotated object boxes and it requires to predict the target object position (heading and elevation in the panorama). The dataset is also split into training, val seen, val unseen and test sets, where

Table 1. Performance on the REVERIE validation unseen and testing split.

	TL	SR↑	OSR↑	SPL↑	RGS↑	RGSPL↑	TL	SR↑	OSR↑	SPL↑	RGS↑	RGSPL↑
Human	-	-	-	-	-	-	21.2	81.5	86.8	53.7	77.8	-
Baseline [16]	45.3	14.4	28.2	7.2	7.8	-	39.1	19.9	30.6	11.6	11.3	6.1
SIA [13]	41.5	31.5	44.7	16.3	22.4	11.6	48.6	30.8	44.6	14.9	19.0	9.2
Rec [11]	16.8	30.7	35.0	24.9	18.8	15.3	15.9	29.6	32.9	24.0	16.5	13.5
Airbert [9]	18.7	27.9	34.5	21.9	18.2	14.2	17.9	30.3	34.2	23.6	16.8	13.3
GMAP	24.9	**40.9**	**43.3**	**28.6**	**29.7**	**20.5**	22.6	**47.4**	**49.7**	**34.9**	**31.1**	**23.1**

Table 2. Performance on the SOON validation unseen and testing split.

	TL	SR↑	OSR↑	SPL↑	RGSPL↑	TL	SR↑	OSR↑	SPL↑	RGSPL↑
Human	-	-	-	-	-	-	90.4	91.4	59.2	51.1
Baseline [22]	29.0	19.5	28.5	13.3	1.2	27.9	12.9	21.5	9.2	0.5
GMAP	32.3	**29.1**	**37.0**	**20.1**	**2.5**	38.2	**28.0**	**35.3**	**18.1**	**2.6**

there are 34 scenes and 2,780 instructions in the train set, 2 scenes and 113 instructions in the val seen set, 5 scenes and 339 instructions in the val unseen set, 14 scenes and 1,411 instructions in the test set. The path length ranges from 2 to 21 viewpoints with 9.5 on average, and the instructions are longer than REVERIE with 47 tokens on average.

3.2. Experimental Setup

The stop policy network is initialized from LXMERT pretrained on image-caption datasets [18]. We use a batch size of 64 and train 40k iterations via AdamW optimizer on 2 Tesla P100 GPUs. The best iteration is selected by the stop accuracy on val unseen split. The navigation policy network uses the same model hyper-parameters as the stop policy network. It is trained with batch size of 8 for 100k iterations on a single Tesla P100 GPU.

REVERIE. Since the instructions in the REVERIE dataset are short and there are groundtruth object annotations in REVERIE, we use the groundtruth object names and room names (obtained from Habitat) to train a speaker model to synthesize new instructions. We generate one instruction for each annotated object in the REVERIE train split and combine the augmented data with the original ones in the stop policy training.

SOON. We observed the object annotations in the SOON dataset are not accurate. For the center point annotation, there are slight shifts from the real object position. Moreover, for the annotated object bounding boxes, the annotations are very noisy (about 50% annotated center point are not even inside of the annotated bounding boxes). Therefore, we automatically cleaned the SOON training split. We use the detected bounding boxes [2] and generate pseudo-labeled object boxes based on semantic and distance matching with the annotated object center point. In this way, we make the SOON task similar to the REVERIE one.

3.3. Results

Table 1 and 2 present the performance of our GMAP approach on the REVERIE[1] and SOON[2] testing split. Firstly, we observe significant gains on the OSR metric which measures how many times the agent has passed by the target viewpoint. It demonstrates that our navigation policy is better at object-oriented exploration in a unseen environments than previous approaches. Secondly, the high SPL score shows that our exploration is also efficient. Finally, we use the quotient of SR divided by OSR to estimate the stop accuracy in predicted paths. Our stop policy achieves 94.5% on val unseen split which improves the previous best approach by 6.8%. However, we observe that the object grounding performance is still far from satisfactory especially on the SOON dataset. This is partly due to the noisy annotations, but more importantly to the insufficient generalization ability for unseen environments.

4. Conclusion

The object-oriented VLN task in ICCV 2021 REVERIE and SOON challenges introduces additional difficulties compared to classical VLN tasks such as R2R or RxR. We propose a novel approach for this task, which disentangles the stop and navigation policy and builds a graphical map on-the-fly to allow more efficient exploration in unseen environments. Our method significantly improves over exist-

[1]https://eval.ai/web/challenges/challenge-page/606/leaderboard/1683

[2]https://eval.ai/web/challenges/challenge-page/1275/leaderboard/3235

ing baselines on both the REVERIE and SOON challenge, i.e. the RGSPL metric increases from 13.5% to 23.1% on REVERIE, and from 0.5% to 2.6% on SOON. However, the performance is still far from perfect, which leaves rooms for further improvements.

References

[1] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018. 1

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, pages 6077–6086, 2018. 2, 3

[3] 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*, pages 3674–3683, 2018. 1

[4] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *NeurIPS*, volume 33, 2020. 1

[5] Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *CVPR*, pages 12538–12547, 2019. 1

[6] Zhiwei Deng, Karthik Narasimhan, and Olga Russakovsky. Evolving graphical planner: Contextual global planning for vision-and-language navigation. In *NeurIPS*, volume 33, 2020. 1

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16$\times$ 16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 2

[8] 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*, pages 3318–3329, 2018. 1

[9] Pierre-Louis Guhur, Makarand Tapaswi, Shizhe Chen, Ivan Laptev, and Cordelia Schmid. Airbert: In-domain pretraining for vision-and-language navigation. In *ICCV*, 2021. 2, 3

[10] Yicong Hong, Cristian Rodriguez, Yuankai Qi, Qi Wu, and Stephen Gould. Language and visual entity relationship graph for agent navigation. In *NeurIPS*, volume 33, pages 7685–7696, 2020. 1

[11] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. A recurrent vision-and-language BERT for navigation. In *CVPR*, 2021. 1, 3

[12] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *EMNLP*, pages 4392–4412, 2020. 1

[13] Xiangru Lin, Guanbin Li, and Yizhou Yu. Scene-intuitive agent for remote embodied visual grounding. In *CVPR*, 2021. 2, 3

[14] 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. 1

[15] Alexander Pashevich, Cordelia Schmid, and Chen Sun. Episodic transformer for vision-and-language navigation. In *ICCV*, 2021. 1

[16] Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. Reverie: Remote embodied visual referring expression in real indoor environments. In *CVPR*, pages 9982–9991, 2020. 1, 3

[17] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *CVPR*, pages 10740–10749, 2020. 1

[18] Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. In *EMNLP*, pages 5103–5114, 2019. 2, 3

[19] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. In *NACAL*, pages 2610–2621, 2019. 1

[20] Hanqing Wang, Wenguan Wang, Wei Liang, Caiming Xiong, and Jianbing Shen. Structured scene memory for vision-language navigation. In *CVPR*, 2021. 1

[21] 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. In *CVPR*, pages 6629–6638, 2019. 1

[22] Fengda Zhu, Xiwen Liang, Yi Zhu, Qizhi Yu, Xiaojun Chang, and Xiaodan Liang. Soon: Scenario oriented object navigation with graph-based exploration. In *CVPR*, pages 12689–12699, 2021. 1, 3