3D Visual Grounding with Graph and Attention

Ekrem Alper Kesen Yoonha Choe
Technical University of Munich

Abstract

In this paper, we improve the previous state-of-the-art method named ScanRefer: 3D Object Localization in RGB-D Scans using Natural Language by adapting graph and attention mechanism. ScanRefer is a neural network architecture that localizes objects in 3D point clouds given natural language descriptions referring to the underlying objects. We improve the object detection module of ScanRefer by substituting VoteNet with Back-tracing Representative Points Network (BRNet). We also propose a method of scene-language understanding and objects relationship understanding by adapting graph neural network, language self-attention, and cross-modal attention mechanism. We show that our model outperforms ScanRefer model on ScanRefer benchmark.

1. Introduction

The task of 3D visual grounding is to locate the target objects by generating corresponding bounding box in 3D space based on natural language query. The latest work ScanRefer [1] is state-of-the-art method for 3D visual grounding task, but there is still room for improving several parts of this architecture. Recently, B. Cheng et al. [2] proposed a new 3D object detection method named BRNet and argued VoteNet [5], which is used as 3D object detection method in ScanRefer, has several problems in terms of performance. We replace VoteNet with BRNet to generate high-quality object proposals and to improve the localization results. ScanRefer also suffers from a limited understanding of relationship between the objects in the scene. We tackle this problem by improving language encoding module and fusion module. In specific, we add self-attention mechanism to the encoding module and replace the fusion module with graph module and cross-attention module, which lead to better understanding of the scene description and relation between the objects.

2. Method

There are five main modules in our model: detection, encoding, graph, cross-modal attention, and localization module (Fig. 1). The input description is tokenized with SpaCy [3] using pretrained GloVE word embeddings [4] and GRU. As shown in Fig. 1, in the encoding module, a self-attention module is added on top of the GRU hidden states H. In the detection module, BRNet is used to output object proposals with 3D bounding boxes from point cloud P. The visual features are concatenated with the language embedding of input description and fed into the graph module and cross-modal attention module to produce the fused features. Finally, a localization module predicts confidence scores for the object proposals.

2.1. Back-tracing Representative Points Network

In our method, VoteNet is replaced with BRNet [2] for improving 3D object detection. BRNet architecture consists of four modules: vote generation and clustering, representative points generation, seed points revisiting, and proposal refinement.

Vote generation and clustering Vote generation and cluster module is similar to VoteNet [5]. Given input point cloud, seed points and features are extracted by PointNet++[6] backbone. Votes are generated with network based voting module and grouped into M point clusters.

Representative points generation The module predicts offset distances $d_i \in \mathbb{R}^6$ in six directions from vote center v_i to object surfaces, and orientation angle $\theta_i \in [0, 2\pi]$. It samples representative points R_i along these directions and the objective of the module is described as follows:

$$L_{rep} = \lambda \left(\frac{1}{M_{pos}} \sum_{i=1}^{M} ||d_i - d_i^*||_p \cdot \mathbb{I}[v_i \, is \, positive] \right) + L_{rep-ang}, \quad (1)$$

where M_{pos} is the number of positive vote centers, d_i^* is ground-truth offset distances, p is smooth-L1 norm, and λ balances the offset and angle terms. Orientation angle loss $L_{rep-ang}$ is similar to heading angle loss in VoteNet [5].

Seed points revisiting Seed points within a radius of 0.2 around representative points R_i are revisited and aggregated using a network similar to PointNet. Aggregated seed features from each representative point are fused into a single feature $g_i \in \mathbb{R}^{128}$ by concatenation.

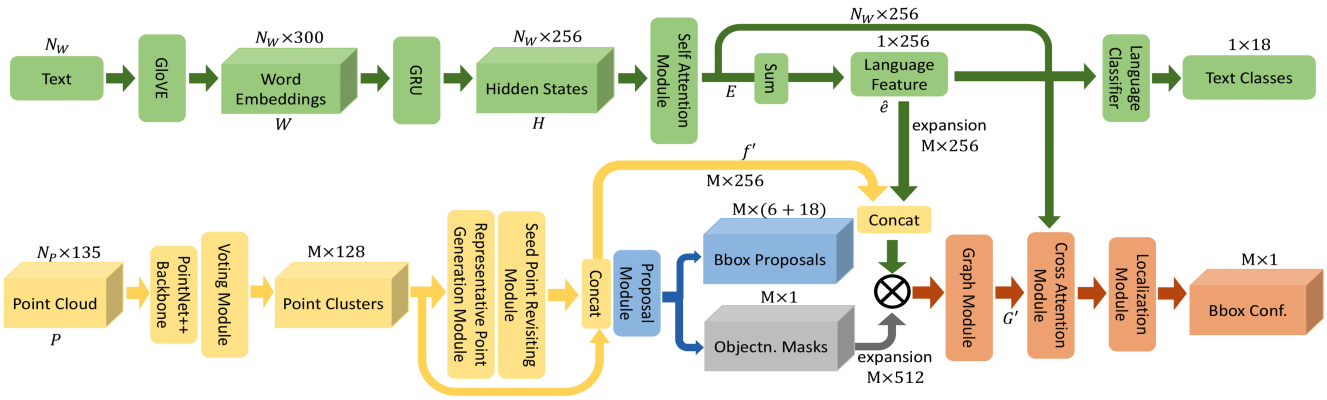

Figure 1. Architecture of ScanRefer with Graph and Attention. PointNet++ [6] backbone is used to extract point feature maps from the preprocessed point cloud P, which are then fused into BRNet [2] to produce object proposals. Object proposals are concatenated with language embedding of input description, which is obtained by pretrained GloVE word embeddings [4], GRU and self-attention module, and masked by the objectness mask, The cross-modal attention module takes the output of graph module and language features as inputs to produce the fused features. Finally, the localization module predicts confidence scores for each object proposal.

Proposal refinement Aggregated revisited seed features g_i and feature vectors f_i are fused as $f'_i = [f_i, g_i] \in \mathbb{R}^{256}$ and fed into a proposal module which predicts the residuals Δd_i and $\Delta\theta_i$. Refined distance and angle are denoted as $d'_i = d_i + \Delta d_i$, $\theta'_i = \theta_i + \Delta\theta_i$. The size and center of the bounding box can be computed from refined distance and vote center. Semantic classification scores are also predicted by this module. The objective of the module is defined as follows:

$$L_{refine} = \frac{1}{M_{pos}} \sum_{i=1}^{M} (\lambda ||d'_i - d^*_i||_p + ||\theta'_i - \theta^*_i||_p) \cdot \mathbb{I}[v_i \; is \; positive], \quad (2)$$

where θ^*_i is the ground truth orientation angle of the bounding box.

2.2. Language Self-Attention Module

In our model, we add self-attention mechanism on top of the GRU hidden states $H \in \mathbb{R}^{N_W \times 256}$ to make the model attend to more important tokens [10]. The key, query, and value in attention mechanism [7] are generated from hidden states H. In formula, by taking every hidden state of each time step of GRU, the attention weight matrix A is generated by

$$A = softmax(HWH^T), \quad (3)$$

where W is the weight matrix which is optimized through backpropogation and the softmax function is applied to normalize the attention weights. Then, hidden state of each timestep is dotted with attention weight matrix A. Finally, the language embedding $\hat{e} \in \mathbb{R}^{1 \times 256}$ is computed by aggregating all weighted hidden states as follows:

$$E = AH, \quad (4)$$

$$\hat{e} = \sum_{i=0}^{N_W} e_i, \quad (5)$$

where e_i is an element of attention value matrix E.

2.3. Graph Module

A graph module is proposed for the relation encoding among the object proposals and their surrounding objects [10]. Concatenated visual and language features are fed into the graph module, which leads to scene-language understanding. We adapt dynamic graph CNN (DGCNN) [8] which is dynamically updated after each layer of the network. In practice, given the feature vector of i-th object proposal f'_i obtained from BRNet, graph module first searches k neighborhoods in feature space that have the closest Euclidean distance to f'_i to construct k-nearest neighbor graph of G. The edge feature e_{ij} is defined as $e_{ij} = h_\theta(f'_i, f'_j)$, where h_θ is non-linear function with learnable parameters θ. We adopt edge function as same as DGCNN paper [8].

$$h_\theta(f'_i, f'_j) = h_\theta(f'_i, f'_j - f'_i). \quad (6)$$

This allows the network to capture global structure of the scene and also local neighborhood information of each object [8]. Finally, edge convolution is applied to aggregate features of all edges emanating from vertex f'_i as follows:

$$g'_i = max_{j:(i,j)\in\epsilon} h_\theta(f'_i, f'_j), \quad (7)$$

where g'_i is the output of edge convolution at i-th object proposal and ϵ are the edges. We choose k as 6 as in Z. Yuan et al. [10] which also adapts DGCNN to improve ScanRefer performance and shows good results. We also add skip connection to prevent small gradients in a deep architecture.

	unique		multiple		overall	
	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5
Baseline (xyz+rgb+lobjcls)	63.47	43.25	30.61	19.36	36.98	23.99
BRNet (xyz+rgb+lobjcls)	66.37	48.02	31.23	20.51	38.05	25.85
Self-max pool (xyz+rgb+lobjcls)	64.12	45.01	29.86	19.08	36.51	24.11
Self-sum (xyz+rgb+lobjcls)	66.34	44.20	30.41	19.19	37.38	24.04
DGCNN-fused/skip (xyz+rgb+lobjcls)	65.86	45.76	29.49	19.05	36.55	24.23
DGCNN-fused/skip + Cross (xyz+rgb+lobjcls)	66.41	46.23	29.97	19.18	37.04	24.43
DGCNN-visual/skip + Cross (xyz+rgb+lobjcls)	63.54	43.30	28.84	18.81	35.77	23.76
DGCNN-fused + Cross (xyz+rgb+lobjcls)	64.94	44.23	30.51	19.64	37.19	24.41
Baseline (xyz+multiview+normals+lobjcls)	76.33	53.51	**32.73**	21.11	41.19	27.40
Ours (xyz+multiview+normals+lobjcls)	**76.91**	**55.43**	32.63	**21.56**	**41.23**	**28.13**
Test results (ScanRefer benchmark)						
Baseline (xyz+multiview+normals+lobjcls)	68.59	43.53	**34.88**	**20.97**	**42.44**	26.03
Ours (xyz+multiview+normals+lobjcls)	**70.16**	**52.02**	32.33	19.59	40.81	**26.86**

Table 1. Comparison of localization results obtained by our ScanRefer with Graph and Attention models and baseline model. Our model calculates the accuracy where the intersection over union (IoU) between the prediction and ground truth is higher than 0.25 and 0.5. Accuracy is reported on "unique" and "multiple" subsets. "Unique" means that there is only a single object of its class in the scene.

ScanNet V2	cab	bed	chair	sofa	tabl	door	wind	bkshf	pic	cntr	desk	curt	fridg	showr	toil	sink	bath	ofurn	mAP
VoteNet	7.5	71.3	61.9	59.2	40.6	12.4	5.2	17.7	0.1	8.4	37.8	4.3	22.6	2.3	67.8	17.9	53.4	5.1	27.5
BRNet	**12.0**	**74.7**	**69.9**	**65.9**	**46.5**	**24.1**	**17.5**	**29.5**	**2.7**	**16.0**	**46.3**	**19.8**	**29.2**	**10.5**	**85.9**	**22.3**	**71.0**	**7.0**	**36.1**

Table 2. 3D Object Detection results on ScanNet V2 validation set with average precision with 3D IoU threhold of 0.50.

2.4. Cross-Modal Attention Module

We propose to use a cross-modal attention mechanism between language and visual features, which allows the model to sufficiently capture the correlations among tokens in the language description and the object proposals. [9]. Given the output of graph module $G' \in \mathbb{R}^{M \times 128}$ and self-attention module $E \in \mathbb{R}^{Nw \times 256}$ before aggregation, it produces a query, key, and value by linear transformations as

$$Q = W_q G', K = W_k E, V = W_v E, \tag{8}$$

where W_q, W_k, W_v are parameters to be learned through backpropogation. We compute cross-modal attention matrix B as follows:

$$B = softmax(QK^T). \tag{9}$$

Each element b_{ij} in matrix B represents how much related the i-th object proposal is with j-th token in the description. Finally, we compute cross-modal attention value matrix C which takes into account all tokens as much as cross-modal attention weights for each object proposal as follows:

$$C = BV. \tag{10}$$

Those attention values are added with the output of graph module again by skip connection and fed into the localization module which predicts the final confidence score for each proposal.

2.5. Loss Function

The final loss is a linear combination of the localization, object detection, and the language to object classification loss, which is similar to ScanRefer [1]: $L = 0.1L_{loc} + 10L_{det} + L_{cls}$. However, since we replace VoteNet with BRNet, the object detection loss is modified accordingly.

$$L_{det} = L_{refine} + L_{rep} + 0.5L_{objn\_cls} + 0.1L_{sem\_cls}, \tag{11}$$

where $L_{obj-cls}$ is the objectness loss and $L_{sem-cls}$ is the semantic classification loss.

3. Experiments

3.1. Quantitative Analysis

We evaluate the performance of our model against baseline which is ScanRefer on the validation set and the test set (ScanRefer benchmark) (see Tab. 1). Note that we evaluate the performance with geometry and RGB values (xyz+rgb+lobjcls) and test the performance with geometry, multi-view image features, and normals (xyz+multiview+normals+lobjcls), which leads to a better

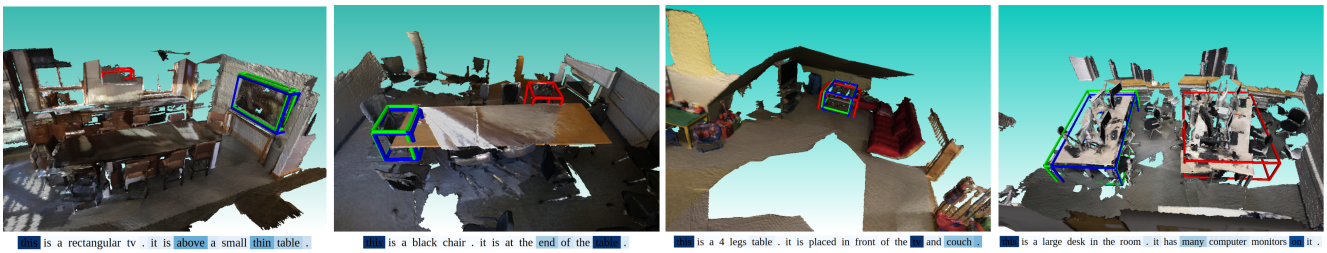

Figure 2. The visualization results from our method, baseline, and ground truth bounding box. Ground truth boxes are marked with green, predicted boxes from baseline are marked with red, and predicted boxes from our method are displayed with blue. The background color of each token in the description represents the self-attention weight.

performance.

First, we train only object detection module with VoteNet and BRNet respectively and compare the object detection results. We use average precision as evaluation metric and compare the performance of detectors with IoU thresholds of 0.50. As shown in Tab. 2, BRNet gives better object detection results compared to VoteNet in all cases with IoU threshold of 0.50. BRNet also leads to higher accuracy of visual grounding task. Our first approach is that only aggregated vote features are fed into the localization module, which achieves Acc@0.25 result of 36.83% and Acc@0.50 result of 25.10%. Another approach is that both aggregated vote features and revisited seed features are fed into the localization module, which produces a better performance (see Tab. 1 BRNet) and also outperforms baseline in all cases.

We show that self-attention module improves the accuracy, especially in unique case. We compare two self-attention methods (see Tab. 1 Self-max pool, Self-sum) to aggregate attention values $E \in \mathbb{R}^{Nw \times 256}$ to $\hat{e} \in \mathbb{R}^{1 \times 256}$. Summation leads to a better performance than max pooling in Acc@0.25 unique and multiple cases.

We also show that DGCNN and cross-modal attention improves the localization results, mainly in unique case. Passing only visual feature into DGCNN and fuse with language feature only in the cross-modal attention module leads to a lower accuracy than passing fused feature. We also add skip connection to DGCNN which leads to a higher accuracy especially in unique case, so we select passing fused feature with skip connection as our final approach.

Our final model which combines BRNet, self-attention, DGCNN, and cross-modal attention module has better performance than baseline except in Acc@0.25 multiple case. We test our final model on ScanRefer benchmark and achieves better result than baseline in unique cases and Acc@0.50 overall.

3.2. Qualitative Analysis

Fig. 2 shows the results provided by baseline, ground truth, and our method. Not only for the unique case but also for the multiple case, we can see our model can predict the corresponding bounding boxes correctly. With the visualization of self-attention weight of the description, we can see the self-attention module helps the model to capture surrounding objects and global localization in the scene.

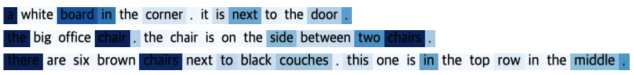

Figure 3. The visualization results of the self-attention module.

We illustrate more visualization results of the self-attention module in Fig. 3. It shows that the self-attention module can capture the features of the object, but also the relationship among the object itself and surrounding objects. For example, as seen in the first sentence, the module recognizes the class of object ('board'), global localization ('corner', 'next'), and the class of the neighborhood object ('door') correctly.

4. Conclusion

In this work, we improve ScanRefer architecture which localizes objects in the 3D scene corresponding to natural language query. By changing its 3D object detection module to BRNet, it benefits from back-tracing strategy, so that it achieves better detection performance than VoteNet. We also utilize self-attention module that allows better understanding of scene descriptions. We show that this module can capture the features of target object and information of its neighboring objects by visualizing attention weights. Furthermore, we adapt graph neural network and cross-modal attention to fuse visual and language features, which improve understanding of neighboring objects in the scene and sufficiently capture correlations between the description and the scene. Our model can be further improved by hyperparameter tuning for self-attention, graph, and cross-modal attention module.

References

[1] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. *16th European Conference on Computer Vision (ECCV)*, 2020.

[2] Bowen Cheng, Lu Sheng, Shaoshuai Shi, and Dong Xu Ming Yang. Back-tracing representative points for voting-based 3d object detection in point clouds. *Computer Vision and Pattern Recognition (CVPR)*, 2021.

[3] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

[4] AJeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[5] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. *International Conference on Computer Vision*, 2019.

[6] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Neural Information Processing Systems*, 2017.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[8] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 2019.

[9] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation, 2019.

[10] Zhihao Yuan, Xu Yan, Yinghong Liao, Ruimao Zhang, Zhen Li, and Shuguang Cui. Instancerefer: Cooperative holistic understanding for visual grounding on point clouds through instance multi-level contextual referring. *arXiv preprint*, 2021.