Relative Position Relationship Learning Network for Scene Graph Generation

Zhi Chen[1,2] and Yibing Zhan[1,2(✉)]

[1] Hangzhou Dianzi University, Hangzhou, China
zhixiao996@hdu.edu.cn
[2] JD Explore Academy, Beijing, China
zhanyibing@jd.com

Abstract. Scene graph generation (SGG) aims to detect objects along with their relationships in images. It is well believed that the position of objects is a significant consideration when analyzing object relationships. However, current SGG methods generally adopted the absolute positions of objects, which are less effective to describe relationships between two objects when the two objects are placed into different positions of one image. In this paper, we propose a relative position relationship learning network (RPRL-Net) to explicitly represent relationships between different positional objects. Specifically, RPRL-Net develops relative positional self-attention (RPSA) modules to analyze context features from objects by exploring relative positional information between pairwise objects. Afterward, RPRL-Net integrates absolute positional features, relative positional features, and context features of object pairs to predict the final predicates. We conducted comprehensive experiments on the Visual Genome dataset. The experimental results compared with the state-of-the-art demonstrate the superiority of RPRL-Net.

Keywords: Relative position · Self-attention · Scene graph generation

1 Introduction

Scene graph generation (SGG) aims to generate scene graphs of images to model objects and their relationships. In the summary graph, the nodes represent detected objects, and the edges represent the relationships between object pairs. Scene graphs have been adopted in a wide range of high-level visual tasks, such as image captioning [1] and visual question answering [2]. Due to the wide application of scene graphs [3–6], SGG has become a hot topic recently.

In scene graph generation, a scene graph is collection of a visual triplets: subject-predicate-object, such as woman-holding-food and man-eating-food, which as shown in Fig. 1. When predicting relationships, one key is to explore and exploit the rich semantic and spatial information of pairwise objects. However, most current SGG methods only exploited visual information, semantic information and absolute positional information [7] of single objects, which can not explicitly and effectively model their

This work is supported by the Major Science and Technology Innovation 2030 "New Generation Artificial Intelligence" key project (No. 2021ZD0111700) and the National Natural Science Foundation of China (Grant No. 62002090).

L. Fang et al. (Eds.): CICAI 2022, LNAI 13604, pp. 551–559, 2022.
https://doi.org/10.1007/978-3-031-20497-5_45

Fig. 1. Examples of different relative positional information represent different relationships

relationships among pairwise objects. It is more significant to model the relative positional information of pairwise objects since different relative positional features may represent different relationships. As shown in Fig. 1 on the left, the food is far away from the woman, so holding is predicted. in Fig. 1 on the right, the food is near to the man, eating better describes their relationship between man and food than holding. Inspired by [8], we model relative positional information between object pairs by using the relative positions, including relative distances, relative scales and relative orientations. Methodologically, most existing approaches model semantic and spatial information by using the CNN framework [9], the RNN framework [10], or the attention framework [11]. Despite the success of these methods, they usually use an iterative modeling strategy to represent the single object context, which may limit the capability of modeling the contextualized representations.

In this paper, we propose a relative position relationship learning network (RPRL-Net) to explore and exploit the relative positional information of pairwise objects for SGG. To overcome the suboptimality of modeling the absolute positional information of single object and the iterative context modeling mechanism, a relative positional self-attention(RPSA) module is proposed to encode the relative positional information into objects and relationship contexts. Besides, in order to facilitate the fusion of semantic and relative positional information, a new technique is developed to encourage increased interaction between query, key and relative position embedding in the RPSA. Finally, we propose positional triplets,i.e., the absolute positional feature of subject and object as well as the relative positional feature between them, respectively. By fusing relationship contexts and positional triplets to predict relationships. The main contributions of this paper lie in two aspects:

- In this paper, we propose a relative position relationship learning network(RPRL-Net) to explicitly represent their relationships between different positional objects for SGG. Besides, a relative positional self-attention(RPSA) module is developed to encode the relative positional information into object and relation contexts.
- We perform extensive experiments on the Visual Genome (VG) dataset [12] and compare RPRL-Net with state-of-the-art scene graph generation methods. Experimental results verify the superior performance of the RPRL-Net compared with the state-of-the-art approaches.

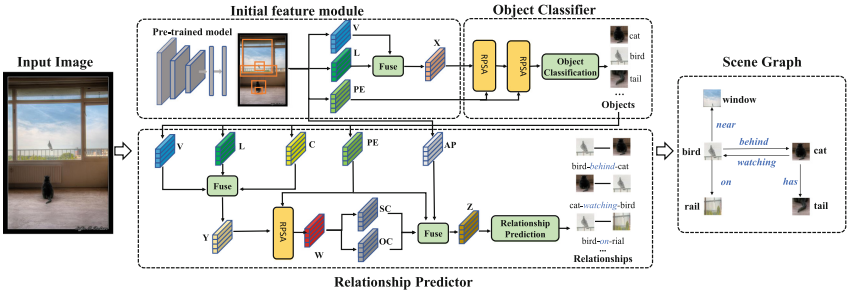

Fig. 2. Flowchart of RPRL-Net consists of three modules: an initial feature module (IFM), an object classifier and a relationship predictor. RPRL-Net first obtains visual features (V), linguistic features (L) and relative positional features (PE) based on the IFM. Then, fusion features (X) and PE are fed to stacked RPSA module to obtain updated context features (C) and predict object label. Afterwards, updated fusion features (Y) and PE are fed to stacked RPSA module to obtain context features (W). The relationship predictor finally predicts the relationships based on updated context features (Z).

The rest of this paper is organized as follows. Section 2 presents the details of our methods. Section 3 presents the experiments, followed by conclusion in Section 4.

2 Approach

In this section, we introduce the architecture of relative position relationship learning network (RPRL-Net) for SGG. Firstly, the feature representations from the input image based on a pre-trained object detector model is described. Then, we explain the details of RPSA module. Finally, the details of object classifier and relationship predictor are explained. An overview flowchart of RPRL-Net is shown in Fig. 2.

2.1 Initial Feature Module

We use Faster R-CNN to detect objects of input images [13]. For one image I, the initial feature module generates four types of features.

Visual Features: Each detected object is represented as a 4096-d vector by extracting fc7 feature after RoI Align and fc6 layer. Finally, the visual features represent $\mathrm{V} \in \mathbb{R}^{m\times 4096}$.

Linguistic Features: We use a pretrained 300-d word embedding model [14] to transform the discrete labels into continuous linguistic features, obtaining a linguistic feature matrix of $\mathrm{L} \in \mathbb{R}^{m\times 300}$.

Absolute Positional Features: Absolute positional feature $\mathrm{AP} \in \mathbb{R}^{m\times 9}$ includes the bounding box $(\frac{x_1}{w}, \frac{y_1}{h}, \frac{x_2}{w}, \frac{y_2}{h})$, center $(\frac{x_1+x_{i2}}{2w}, \frac{y_1+y_2}{2h})$, sizes $(\frac{x_2-x_1}{w}, \frac{y_2-y_1}{h}, \frac{(x_2-x_1)(y_2-y_1)}{wh})$. Here, (x_1, y_1, x_2, y_2) are the bounding box coordinates of object proposals B. w and h are the image width and height.

Relative Positional Features: For m-th object and n-th object, the relative distances are calculated as:

$$\boldsymbol{d}_{mn} = [\log(\frac{|x_m - x_n|}{w_m}), \log(\frac{|y_m - y_n|}{h_m})] \tag{1}$$

the relative scales are defined as:

$$\boldsymbol{s}_{mn} = [\log(\frac{|w_n|}{w_m}), \log(\frac{|h_n|}{h_m})] \tag{2}$$

and the relative orientation is calculated as a cosine function:

$$\boldsymbol{o}_{mn} = \frac{x_m - x_n}{\sqrt{(x_m - x_n)^2 + (y_m - y_n)^2}} \tag{3}$$

Finally, the relative positional features are represented as:

$$\boldsymbol{pos}_{mn} = [\boldsymbol{d}_{mn}, \boldsymbol{s}_{mn}, \boldsymbol{o}_{mn}] \tag{4}$$

This 5-d relative positional features are embedded to a high-dimensional representation by method in [15], which computes cosine and sine functions of different wavelengths.

$$\mathrm{PE}_{(i,pos)} = (\sin(\mathrm{pos}/1000^{2i/d_{model}}) || \cos(\mathrm{pos}/1000^{2i/d_{model}})) \tag{5}$$

where pos is the relative position and i is the dimension. That is, each dimension of the positional encoding corresponds to a sinusoid. d_{model} is the dimension of output feature. Finally, we obtain a feature matrix of PE $\in \mathbb{R}^{m \times n \times 64}$.

Fusion Features: We concatenate V and L are concatenated together and then linearly transform them to a matched dimensionality, resulting in the fused features X $\in \mathbb{R}^{m \times 1024}$. The process is calculated as:

$$\mathrm{X} = \mathrm{Linear}(\mathrm{V} \| \mathrm{L}) \tag{6}$$

2.2 Relative Position Self-attention Module

Let X $\in \mathbb{R}^{N \times d_o}$ denote the fusion feature set of objects. d_o is the feature dimension of X. X is first fed into three parallel linear layers to obtain the queries Q, keys K, and values V, respectively. Q, K, V is defined as:

$$\mathrm{Q} = \mathrm{Linear}(\mathrm{X}), \mathrm{K} = \mathrm{Linear}(\mathrm{X}), \mathrm{V} = \mathrm{Linear}(\mathrm{X}) \tag{7}$$

where Q, K ,V $\in \mathbb{R}^{m \times d_k}$, d_k is output feature dimension. Original self-attention module uses a scaled dot-product, which represents to compute similarity of fusion features. Inspired by [8], we encode the relative positional features into fusion features. The self-attention mechanism be rewritten as:

$$\mathrm{SA}(\mathrm{Q}, \mathrm{K}, \mathrm{V}, \mathrm{RP}) = \mathrm{Softmax}(\frac{\mathrm{QK}^T}{\sqrt{d_k}} + \mathrm{RP})\mathrm{V} \tag{8}$$

where $\sqrt{d_k}$ is a scaling factor following [15]. RP $\in \mathbb{R}^{m \times 64}$ is the updated relative positional feature. RP is defined as:

$$\mathrm{RP} = \mathrm{FC}(\mathrm{Q} + \mathrm{K} + \mathrm{FC}(\mathrm{PE})) \tag{9}$$

where FC (PE) corresponds to a linear layer applied to the last axis of PE. RP is the sum of query, key and relative position feature, which increases the interaction among them. In this method, RP serves as a gate to filter out the dot product of query and key. This gate would prevent a query from attending to a similar key (content-wise) heavily if the query and key positions are far away from each other.

The multihead variant of the attention module is popularly used which allows the model to jointly attend to information from different representation sub-spaces, and is defined as

$$\text{Multi-Head}(\mathrm{Q}, \mathrm{K}, \mathrm{V}, \mathrm{RP}) = \mathrm{Concat}(\mathrm{head}_1, \cdots, \mathrm{head}_H)\mathrm{W}^o \tag{10}$$

$$\mathrm{head}_k = \mathrm{SA}(\mathrm{Q}, \mathrm{K}, \mathrm{V}, \mathrm{RP}) \tag{11}$$

Finally, we further combine with the FFN layer to generate the relative positional self-attention module, which contains two fully connected layers:

$$\mathrm{FFN}(X) = \mathrm{FC}_{o1}(\sigma(\mathrm{FC}_{o2}(X))) \tag{12}$$

where σ indicates ReLU. The residual connection with layer normalization [15], which is defined as X = X + LN(Fun(X)), is added to each attention network and each FFN. Here, X is the input feature set, LN$(\cdot)$ indicates layer normalization, and Fun$(\cdot)$ represents either an attention network or a FFN.

2.3 Object Classifier

In object classification, with considering the relative positional information and the interaction among the key, query, and relative position embedding, the fusion features and relative positional features are fed into stacked RPSA module to obtain the object context features. Then, the object context features X are projected into c-dimensional vectors $O \in \mathbb{R}^{m \times c}$, where c is the number of object classes. Finally, we predict the refined object labels by using a softmax cross-entropy loss based on the c-dimensional vectors.

2.4 Relationship Predictor

Suppose an object proposal set $\mathcal{B} = \{b\}$ is given. The updated fusion features Y of object proposals $\mathcal{B}$ is initialized by fusing the visual features, linguistic features obtained from the corresponding object proposals $\mathcal{B}$ and the object context features obtained from the last RPSA module layer. Y of $\mathcal{B}$ is calculated as:

$$\mathrm{Y} = \sigma(\mathrm{FC}(\mathrm{V} \| \mathrm{L} \| \mathrm{C})) \tag{13}$$

Then, we feed Y into stacked RPSA module to obtain subject context features of subject proposals SC and object context features OC of object proposals, respectively.

Afterwards, the edge context features Z_{so} between object pairs v_{so} is calculated as:

$$\mathrm{Z}_{so} = \sigma(\mathrm{FC}_{v3}(\mathrm{FC}_{v1}(\mathrm{SC}) \| \mathrm{FC}_{v2}(\mathrm{OC}) \| \mathrm{FC}(\mathrm{PE}_{so} + \mathrm{AP}_s + \mathrm{AP}_o))) \tag{14}$$

Table 1. Performance comparison on SGDet of the VG dataset. We compute the R@20, R@50, R@100 and their mean with and without Graph constrained. "–" indicates the results are unavailable. The best performance is highlighted in boldface.

Method	With graph constraint				Without graph constraint			
	R@20	R@50	R@100	Mean	R@20	R@50	R@100	Mean
IMP [16,17]	18.1	25.9	31.2	25.1	18.4	27.0	33.9	26.4
VtransE [12,17]	23.1	29.9	34.7	29.2	24.4	33.1	39.8	32.4
Motif [17,18]	25.5	32.8	37.2	31.8	27.0	36.6	43.4	35.7
VCTree [17,19]	24.5	31.9	36.2	30.9	26.1	35.7	42.3	34.7
Motif-cKD [20]	25.2	32.5	37.1	31.6	–	36.3	43.2	–
VCTree-cKD [20]	24.8	32.0	36.1	31.0	–	35.9	42.4	–
RPRL-Net	**25.6**	**33.1**	**37.6**	**32.1**	**27.3**	**37.2**	**44.1**	**36.2**

where the AP_s-PE_{so}-AP_o indicates a position triplets, which consists of the absolute positional features of subject and object as well as the relative positional features between subject and object, respectively. Finally, we use the binary cross-entropy loss predict the relationship labels.

3 Experiments

In this section, we conduct experiments to verify the effectiveness of the RPRL-Net on the commonly used benchmark.

3.1 Experimental Settings

Dataset: We use the Visual Genome dataset [12] to conduct all experiments. The VG dataset contains 108,077 images with average annotations of 38 objects and 22 relations per image. Following previous works in [17,18], the most frequent 150 object categories and 50 predicate categories are utilized for evaluation, which split the dataset into 70K/5K/32K as train/validation/test sets.

Evaluation Tasks and Metrics: Following [17], scene graph detection (SGDet) task for SGG is adopted. SGDet generates scene graphs of images to predict the label of objects and relationships without extra-label information. Recall@K (R@K) is calculated by averaging the recall of the top K relationships of all images [21]. We use recall as the evaluation metric and $K = \{20, 50, 100\}$ is reported in our experiments. The performance with and without graph constraint [18] is considered.

3.2 Implementation Details

To ensure a fair comparison of previous SGG, we use the codebase and pre-trained object detection model provided by [17]. The backbone is the Faster R-CNN with

Table 2. Ablation studies

Method	With constraint		Without constraint	
	R@50	R@100	R@50	R@100
Baseline	32.4	36.8	36.4	43.1
B+SA	32.5	37.0	36.6	43.4
B+SA+P	32.8	37.4	36.9	43.8
B+O-RPSA	33.0	37.5	37.1	43.9
RPRL-Net	**33.1**	**37.6**	**37.2**	**44.1**

ResNeXt-101-FPN [22]. The hyperparameters mostly followed [17]. The SGD optimizer with a momentum of 0.9 is adopted. The warm-up strategy [15] is used to increase the learning rate from 0 to 0.001 in the first 5000 iterations. Then, the learning rate is decayed by 0.1 at 18,000 and 24,000 iterations. All training last for 30,000 iterations. The base learning rate is set to 0.001 and the batch size is set to 12. For each image, the top-80 object proposals are provided, and 256 relationship proposals, we set background/foreground ratio for relationship detection as 3/1.

3.3 Performance Comparison

Table 1 presents the results of RPRL-Net and six SGG methods on SGDet of the VG dataset. The results with and without graph constraints are provided. The best performance is highlighted in boldface. From Table 1, we use the same detector and backbone to extract object features. Compared with the second-best methods, RPRL-Net obtains a gain of 0.9% and 1.6% on R@50, and 1.0% and 1.6% on R@100 with and without graph constraint. RPRL-Net consistently outperforms existing state-of-the-art approaches in terms of R@20, R@50 and R@100 metrics on SGDet. These improvements again reveal the ability of RPRL-Net.

3.4 Ablation Studies

A number of experiments are conducted to explore the reasons behind RPRL-Net's success. The results are shown in Table 2 and discussed below. We design four types of variants with different combinations: **Baseline** does not use RPSA module in object classifier and relationship predictor. **B+SA** use self-attention module without relative positional feature and interaction of Q, K and relative positional feature. **B+SA+P** use self-attention module with relative positional feature but does not use interaction of Q, K and relative positional feature. **B+O-PRSA** represent RPSA module is used in object classifier but not used in relationship predictor.

From Table 2, **B+SA** outperforms **Baseline** and **B+SA+P** outperforms **B+SA**. These improvements validate that relative positional information and the interaction of Q, K, relative positional feature have a positive influence on SGG. **RPRL-Net** outperforming **B+O-PRSA** indicate that using RPSA module both in object classifier and relationship predictor is better than only in object classifier.

4 Conclusion

In this paper, we propose a relative position relationship learning network (RPRL-Net) for SGG to explicitly represent their relationships between different positional objects because of the suboptimality of absolute position. The core of RPRL-Net is the relative positional self-attention (RPSA) module to encode the relative positional information into object and relation context. Moreover, the interaction of context feature as well as Q, K and relative positional feature is proposed to facilitate the understanding of object and relation semantics. Comprehensive experiments are conducted on the VG dataset. The experimental results demonstrate that RPRL-Net has high reasoning and integrating abilities.

References

1. Yao, T., Pan, Y., Li, Y., Mei, T.: Exploring visual relationship for image captioning. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. LNCS, vol. 11218, pp. 711–727. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01264-9_42
2. Teney, D., Liu, L., van Den Hengel, A.: Graph-structured representations for visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2017)
3. Lin, X., Ding, C., Zhan, Y., Li, Z., Tao, D.: HL-Net: heterophily learning network for scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19476–19485 (2022)
4. Zhan, Y., Jun, Yu., Ting, Yu., Tao, D.: Multi-task compositional network for visual relationship detection. Int. J. Comput. Vis. **128**(8), 2146–2165 (2020)
5. Chen, C., Zhan, Y., Yu, B., Liu, L., Luo, Y., Du, B.: Resistance training using prior bias: toward unbiased scene graph generation. arXiv preprint arXiv:2201.06794 (2022)
6. Lin, X., Ding, C., Zhang, J., Zhan, Y., Tao, D.: RU-Net: regularized unrolling network for scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19457–19466 (2022)
7. Zellers, R., Yatskar, M., Thomson, S., Choi, Y.: Neural motifs: scene graph parsing with global context (2017)
8. Hu, H., Gu, J., Zhang, Z., Dai, J., Wei, Y.: Relation networks for object detection. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
9. Yang, J., Lu, J., Lee, S., Batra, D., Parikh, D.: Graph R-CNN for scene graph generation (2018)
10. Xu, D., Zhu, Y., Choy, C.B., Fei-Fei, L.: Scene graph generation by iterative message passing (2017)
11. Qi, M., Li, W., Yang, Z., Wang, Y., Luo, J.: Attentive relational networks for mapping images to scene graphs. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
12. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Fei-Fei, L.: Visual genome: connecting language and vision using crowdsourced dense image annotations. Int. J. Comput. Vis. **123**(1), 32–73 (2017)
13. Ren, G., et al.: Scene graph generation with hierarchical context. IEEE Trans. Neural Netw. Learn. Syst. (2020)
14. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: Conference on Empirical Methods in Natural Language Processing (2014)

15. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
16. Li, Y., Ouyang, W., Zhou, B., Wang, K., Wang, X.: Scene graph generation from objects, phrases and region captions. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1261–1270 (2017)
17. Tang, K., Niu, Y., Huang, J., Shi, J., Zhang, H.: Unbiased scene graph generation from biased training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3716–3725 (2020)
18. Zellers, R., Yatskar, M., Thomson, S., Choi, Y.: Neural motifs: scene graph parsing with global context. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5831–5840 (2018)
19. Tang, K., Zhang, H., Wu, B., Luo, W., Liu, W.: Learning to compose dynamic tree structures for visual contexts. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6619–6628 (2019)
20. Wang, T.-J.J., Pehlivan, S., Laaksonen, J.: Tackling the unannotated: scene graph generation with bias-reduced models. arXiv preprint arXiv:2008.07832 (2020)
21. Xu, D., Zhu, Y., Choy, C.B., Fei-Fei, L.: Scene graph generation by iterative message passing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5410–5419 (2017)
22. Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)