SGTR: GENERATING SCENE GRAPH BY LEARNING COMPOSITIONAL TRIPLETS WITH TRANSFORMER

Anonymous authors
Paper under double-blind review

1 SUPPLEMENTARY

1.1 QUERY REFINEMENT

After each layer of decoder structure, the triplet queries are updated with the decoder output. We design the fusing process that aggregate the representation between the two branches, to further improve the reprensetation of queries. For relationship predicate queries of next layer $\boldsymbol{Q}_{tp}^{t+1}$, we fuse the triplet entities hidden state with the pair-wise fusion function proposed in Zhang et al. (2018), which has been adopted in SGG task. The triplets entities queries $\boldsymbol{Q}_{to}^{l+1}, \boldsymbol{Q}_{ts}^{l+1}$ are updated with the triplet predicate decoder output $\boldsymbol{Q}_{tp}$, with the nears neighbors feature representation on encoder memory $\boldsymbol{Z}^t$, according to the center coordinates of predicted entities position $[x_c, y_c]$.

$$\boldsymbol{Q}_{tp}^{l+1} = \boldsymbol{Q}_{tp}^{l} + \text{ReLU}(\boldsymbol{W}_x \boldsymbol{Q}_{ts}^{l} + \boldsymbol{W}_y \boldsymbol{Q}_{to}^{l}) - ||\boldsymbol{Q}_{ts}^{l} - \boldsymbol{Q}_{to}^{l}||_2^2 \tag{1}$$

$$\boldsymbol{Q}_{ts}^{l+1} = \boldsymbol{Q}_{ts}^{l} + \boldsymbol{Z}^t(m,n) + \text{ReLU}(\boldsymbol{W}_{es} \boldsymbol{Q}_{tp}^{l}) \tag{2}$$

$$\boldsymbol{Q}_{to}^{l+1} = \boldsymbol{Q}_{to}^{l} + \boldsymbol{Z}^t(m,n) + \text{ReLU}(\boldsymbol{W}_{eo} \boldsymbol{Q}_{tp}^{l}) \tag{3}$$

$$m, n \leftarrow \underset{m,n \in [0,1]\times[0,1]}{\arg\min} (||[x_c, y_c] - [m, n]||_1) \tag{4}$$

1.2 MATCHING QUALITY CALCULATION FOR GRAPH ASSEMBLING

We take the matching quality of subject entities and predicates as example. For each factors of distance function is determined by the semantic outputs of entity detector and triplet decoder. The $d_{giou} \in \mathbb{R}^{N_r \times N_e}$, $d_{cos} \in \mathbb{R}^{N_r \times N_e}$, $d_{center} \in \mathbb{R}^{N_r \times N_e}$ are calculated by following process.

$$\boldsymbol{M}^s = d_{loc}(\boldsymbol{B}_s, \boldsymbol{B}_e) \cdot d_{cls}(\boldsymbol{P}_s, \boldsymbol{P}_e), \quad d_{loc}(\boldsymbol{B}_s, \boldsymbol{B}_e) = \frac{d_{giou}}{d_{center}} \tag{5}$$

$$d_{giou} = \max(\min(\text{GIOU}(\boldsymbol{b}_s, \boldsymbol{b}_e), 0), 1), \quad d_{cos} = \frac{\boldsymbol{p}_s \cdot \boldsymbol{p}_e^{\intercal}}{||\boldsymbol{p}_s|| \cdot ||\boldsymbol{p}_e||} \tag{6}$$

$$d_{center}(i,j) = ||[x_c, y_c]_i^s - [x_c, y_c]_i^e||_1 \tag{7}$$

where $[x_c, y_c]^s$ are the center coordinates of one box in $\boldsymbol{B}_s$.

1.3 TRIPLETS MATCHING COST

The triplets predication of the model is $\mathcal{T} = \{(\boldsymbol{b}_e^s, \boldsymbol{p}_e^s, \boldsymbol{b}_e^o, \boldsymbol{p}_e^o, \boldsymbol{p}_p, \boldsymbol{b}_p)\}$. The triplets matching cost $\mathcal{C} \in \mathbb{R}^{N_r \times N_{gt}}$ is composed by three part: predicate cost $\mathcal{C}_p$ and entity cost $\mathcal{C}_e$.

$$\mathcal{C} = \lambda_p \mathcal{C}_p + \lambda_e \mathcal{C}_e \quad \boldsymbol{I}^{tri} = \underset{\mathcal{T}, \mathcal{T}^{gt}}{\arg\min} \mathcal{C} \tag{8}$$

For the predicate cost $\mathcal{C}_p(i,j)$ between the i-th predicates prediction and j-th ground-truth relationship, it is computed according the predicate classification distribution.

$$\mathcal{C}_p(i,j) = \exp\left(\frac{\boldsymbol{p}_{p,i} \cdot \text{one-hot}(p_{p,j}^{gt})}{||\boldsymbol{p}_{p,i}|| \cdot ||\text{one-hot}(p_{p,j}^{gt})||} - 1\right) + \|\boldsymbol{b}_{p,i} - \boldsymbol{b}_{p,j}^{gt}\|_1 \tag{9}$$

where $\boldsymbol{p}_{p,i}$ is the i-th $\boldsymbol{p}_p$ of $\mathcal{T}$, and $p^{gt}_{p,j}$ is the predicate label of the j-th triplet in ground truth. Similarly, $\boldsymbol{b}_{p,i}$ is the i-th center coordinates(subject and object) $\boldsymbol{b}_p$ of the triplet prediction, $\boldsymbol{b}^{gt}_{p,j}$ is entity centers set of the j-th triplet in ground truth.

In entity cost $\mathcal{C}_e(i,j)$ between the i-th triplets prediction and j-th ground-truth relationship,, the calculation is given by:

$$\mathcal{C}_e(i,j) = + w_{giou} \cdot \prod_{\star=\{s,o\}} \exp\left(\max(\min(\text{GIOU}(\boldsymbol{b}^{\star}_{e,i}, \boldsymbol{b}^{\star}_{gt,j}), 0), 1)\right) \tag{10}$$

$$+ w_{l1} \cdot \sum_{\star=\{s,o\}} ||\boldsymbol{b}^{\star}_{e,i}, \boldsymbol{b}^{\star}_{gt,j}||_1 \tag{11}$$

$$+ w_{cls} \cdot \prod_{\star=\{s,o\}} \exp\left(\frac{\boldsymbol{p}^{\star}_{e,i} \cdot \text{one-hot}(p^{(*,gt)}_{e,j})}{||\boldsymbol{p}^{\star}_{e,i}|| \cdot ||\text{one-hot}(p^{(*,gt)}_{e,j})||} - 1\right) \tag{12}$$

where $\boldsymbol{b}^{\star}_{e,i}$ is the i-th entity box location come from the triplets prediction $\mathcal{T}$ after graph assembling, $\boldsymbol{p}^{\star}_{e,i}$ is the i-th entity classification prediction. $\boldsymbol{b}^{\star}_{gt,j}$ is the box location of j-th subject/object in the ground truth triplets, and $p^{(*,gt)}_{e,j}$ is entity(subject/object) class label of j-th ground truth triplets.

Based is cost function, we can obtain the matching of relationship prediction. We adopt the one-to-one Hungarian algorithm into an iterative many-to-one matching. Due to the label efficiency, the relationships can not be exhausted labeled in datasets. The one-to-one matching may lead to unstable training because many foreground relationships will be ignored. The model can not learn the proper NMS mechanism for prediction calibration. To circumvent this, we relax the matching threshold to prevent the NMS mechanism from learning. We iteratively execute T times of Hungarian minimum-cost bipartite graph matching.

1.4 Datasets and Implementation Details

1.4.1 Datasets and Metrics

Visual Genome Datasets For Visual Genome (Krishna et al., 2017) dataset, we take the same split protocol as Xu et al. (2017); Zellers et al. (2018). The most frequent 150 object categories and 50 predicates are adopted for evaluation. To demonstrate the long-tailed recognition performance on VG dataset, we follow the protocol from Li et al. (2021) by dividing the categories into three disjoint groups. We we adopt the evaluation metric **recall@K(R@K)** and **mean recall@K (mR@K)** of SGDet, and also report the **mR@100 on each long-tail category groups**: *head*, *body* and *tail*.

Openimage V6 Datasets The Openimage datasets (Kuznetsova et al., 2020) are large scale vision recognition datasets proposed by Google, and been used as SGG benchmarks in Zhang et al. (2019); Lin et al. (2020); Li et al. (2021); Teng & Wang (2021). We adopt the same data splits with the Li et al. (2021), which has 126,368 images used for training, 1813 and 5322 images for validation and test, respectively, with 301 object categories and 31 predicate categories.

The the weighted evaluation metrics (e.g. wmAP_{phr}, wmAP_{rel}, score_{wtd}) used in previous works (Zhang et al., 2019; Lin et al., 2020; Li et al., 2021; Teng & Wang, 2021). However, we argue that weighted scores are unfair when used to evaluate rare categories. Because it re-weights by multiplying the frequency of categories on per-class performance, low-frequency categories are disregarded, resulting in class unbalanced assessment metrics, even though this metric is more numerically stable, as cited in Zhang et al. (2019). In this work, we will report both weighted and initial performance (e.g. mAP_{phr}, mAP_{rel}, score) in our experiments, for more fair class balance evaluation metrics.

1.4.2 Implementation Details

We use the ResNet-101 and DETR (Carion et al., 2020) as backbone networks and entities detectors, with six layers encoder and six layers decoder. The $N_e = 100$ entities queries with $d = 256$ hidden dimension are used as the proposals for feature aggregation. The same DETR detector parameters are use for all one-stage methods reproduced by us. In our triplets constructor, we use the 3 layers

encoders. In triplets decoder, we adopt 12 layers decoder for predicate branch, and 5 layers decoder for entity branch, with $N_r = 150$ queries with $d = 256$ hidden dimensions. For two-stage methods, we use the Faster-RCNN detector with the ResNet-101 backbone.

To speedup the convergence, we first train the entities detector on the target dataset. Then, using this pre-trained detector, we train the relationship detector parts. The key difference between this work and previous work Kim et al. (2021); Li et al. (2021) is that we do not need to fix the parameters of the entities detector to avoid performance drop in SGG training. We keep the parameters of detector in training mode, that still can preserve the considerable performance, or obtain better performance in SGG training.

REFERENCES

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pp. 213–229. Springer, 2020.

Bumsoo Kim, Junhyun Lee, Jaewoo Kang, Eun-Sol Kim, and Hyunwoo J Kim. Hotr: End-to-end human-object interaction detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 74–83, 2021.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision(IJCV)*, 2020.

Rongjie Li, Songyang Zhang, Bo Wan, and Xuming He. Bipartite graph network with adaptive message passing for unbiased scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11109–11119, 2021.

Xin Lin, Changxing Ding, Jinquan Zeng, and Dacheng Tao. Gps-net: Graph property sensing network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3746–3753, 2020.

Yao Teng and Limin Wang. Structured sparse r-cnn for direct scene graph generation. *arXiv preprint arXiv:2106.10815*, 2021.

Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 5410–5419, 2017.

Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5831–5840, 2018.

Ji Zhang, Kevin J. Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical Contrastive Losses for Scene Graph Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2019.

Yan Zhang, Jonathon Hare, and Adam Prügel-Bennett. Learning to count objects in natural images for visual question answering. *arXiv preprint arXiv:1802.05766*, 2018.