Knowledge-Embedded Routing Network for Scene Graph Generation

Motivation

深入理解一个场景，除了要能够定位和识别外，还需要推断图片中目标之间的关系，现实世界中**关系的分布**是不均衡的，现有方法在频率小的关系上表现很差。

作者发现目标对及其关系之间的统计相关性可以有效地调整语义空间，从而很好地解决分布不平衡的问题。

作者将这些统计相关性加入到深度神经网络中，通过开发一个Knowledge-Embedded Routing Network（KERN）来促进场景图的生成。

更具体地说，图像中出现的对象及其关系之间的统计相关性，可以用一个结构化的知识图来明确表示，并学习了一种路由机制来通过图进行消息传播，以探索它们之间的交互作用。

方法

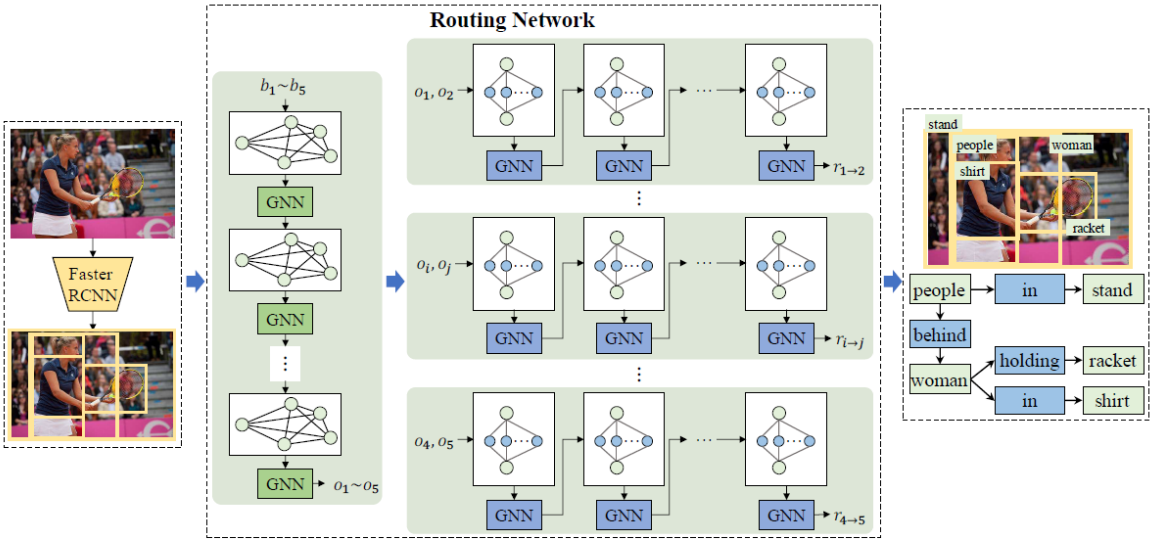


Figure 3. An overall pipeline of the knowledge-embedded routing network. Given an image, we first adopt the Faster RCNN to detect a set of regions. Then, a graph is built to correlate the regions, and a graph neural network is employed to learn contextualized representation to predict the class label for each region. For each object pair with predicted labels, we build another graph to correlate the given object pair with all the possible relationships and employ a graph neural network to infer their relationship. The process is repeated for all object pairs and the scene graph is generated.

场景图定义 $\mathcal{G} = \{B, O, R\}$

$B = \{b_1, b_2, \ldots, b_n\}$ ， $b_i \in R^4$ 表示第i个区域的坐标，

$O = \{o_1, o_2, \ldots, o_n\}$ ， $o_i \in N$ 表示区域i对应的标签，

$R = \{r_{1\to 2}, r_{1\to 3}, \ldots, r_{n\to n-1}\}$ ， $r_{i\to j}$ 是一个三元组，包括一个主体 $(b_i, o_i) \in B \times O$，客体 $(b_i, o_i) \in B \times O$，以及一个关系标签 $x_{i\to j} \in \mathcal{R}$ 。

给定一张图片 $\mathcal{I}$，scene graph的分布可以分解为三个组件

$$p(\mathcal{G}|I) = p(B|I)p(O|B,I)p(R|O,B,I).$$

Bounding box localization

对每张图片使用Faster RCNN生成区域集合 $B = \{b_1, b_2, \ldots, b_n\}$ ， $b_i \in R^4$ ，使用ROI pooling层获取区域特征向量 $\mathbf{f}_i$ 。

Knowledge-embedded routing network

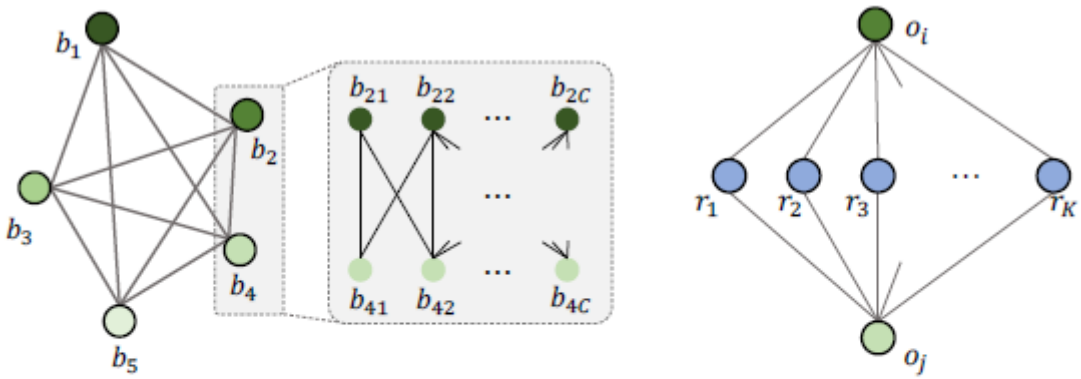


Figure 2. (a) A graph correlating the detected regions appearing in an image; (b) A graph correlating given object pair o_i and o_j with all the relationships.

Object

首先对目标数据集的训练集上不同类别对象的共现概率进行统计。

对于两个类别 c, c'，在c'出现的情况下c出现的概率记为 $m_{cc'}$ ，得到所有类的co-occurrence概率矩阵 $M_c \in \mathbb{R}^{C\times C}$。

给定两个区域b_i和b_j，将b_i复制C次得到C个节点，其中$b_i c$表示b_i和类别C的关系，在b_j上也进行相同的操作。然后用矩阵M_c来关联区域b_i和b_j的节点。

用这样的方法吧所有的区域的关联起来，构建一个图。

采用gated recurrent update机制来迭代地传送信息，

在时间t，每个结点b_{ic}有一个隐藏状态$\mathbf{h}_{ic}^t$，使用结点关联区域的特征向量来初始化隐层状态

$h_{ic}^0 = \varphi_o(f_i)$ ， φ_o 将 h_{ic}^t 映射到一个低维向量。

在每个时间步t，每个结点根据图的结构聚合邻域信息

$$\mathbf{a}_{ic}^{t}=\left[\sum_{j=1,j\neq i}^{n}\sum_{c'=1}^{C}m_{c'c}\mathbf{h}_{jc'}^{t-1},\sum_{j=1,j\neq i}^{n}\sum_{c'=1}^{C}m_{cc'}\mathbf{h}_{jc'}^{t-1}\right]$$

然后模型将 a_{ic}^{t} 和上一步的隐藏状态作为输入，通过门控机制更新隐藏状态

$$\begin{aligned}\mathbf{z}_{ic}^{t}&=\sigma(\mathbf{W}_{o}^{z}\mathbf{a}_{ic}^{t}+\mathbf{U}_{o}^{z}\mathbf{h}_{ic}^{t-1})\\\mathbf{r}_{ic}^{t}&=\sigma(\mathbf{W}_{o}^{r}\mathbf{a}_{ic}^{t}+\mathbf{U}_{o}^{r}\mathbf{h}_{ic}^{t-1})\\\widetilde{\mathbf{h}_{ic}^{t}}&=\tanh\left(\mathbf{W}_{o}\mathbf{a}_{ic}^{t}+\mathbf{U}_{o}(\mathbf{r}_{ic}^{t}\odot\mathbf{h}_{ic}^{t-1})\right)\\\mathbf{h}_{ic}^{t}&=(1-\mathbf{z}_{ic}^{t})\odot\mathbf{h}_{ic}^{t-1}+\mathbf{z}_{ic}^{t}\odot\widetilde{\mathbf{h}_{ic}^{t}}\end{aligned}$$

经过 T_o 次更新，结点信息在图上充分传播，获得每个区域最终的隐状态

$$\{h_{i1}^{T0},h_{i2}^{T0},\ldots,h_{iC}^{T0}\}$$

根据初始状态和最终的隐状态计算每个节点的输出特征

$f_{ic}^{o}=o_{o}(h_{ic}^{0},h_{ic}^{T_o})$ ，o_o 用全连接层实现。

最后对于每个区域，聚集所有的相关的输出特征向量来预测class label

$$o_i=\phi_o(f_{i1}^{0},f_{i2}^{0},\ldots,f_{iC}^{0})$$

输出的类别标签$o_i=argmax(\mathbf{o}_i)$用来做关系推断。

Relationship

作者将目标对和他们的关系的关联以structured graph形式表示，并采用另一个图神经网络来探索这两个部分的相互作用，进行最后的关系推断。

给定一个主体类别c和 一个客体类别c'，先计算所有可能的关系的概率，得到 $\{m_{cc'1},m_{cc'1},\ldots,m_{cc'K}\}$ ，K表示关系数目。

对于o_i 和 o_j ,构建包含一个subject node，一个object node，K个关系结点的图。

$m_{o_io_jk}$ 表示 o_i 和关系节点k的关联，以及 o_j 和关系节点k的关联。

每个结点 $v\in V=\{o_i,o_j,1,2,\ldots,K\}$ 有一个隐藏态 h_v^t。

用目标结点对应区域的特征向量初始化**目标结点**，用两个目标的并集区域的特征向量以及两个目标的空间关系来初始化**关系结点**

$$\mathbf{h}_v^0 = \begin{cases} \varphi_{o'}(\mathbf{f}_i) & \text{if } v \text{ is the object node } o_i \\ \varphi_r(\mathbf{f}_{ij}) & \text{if } v \text{ is a relationship node} \end{cases},$$

在每个时间步t，每个节点聚集其他节点的信息：

$$\mathbf{a}_v^t = \begin{cases} \sum_{k=1}^{K} m_{o_i o_j k} \mathbf{h}_k^{t-1} & \text{if } v \text{ is a object node} \\ m_{o_i o_j k}(\mathbf{h}_{o_i}^{t-1} + \mathbf{h}_{o_j}^{t-1}) & \text{if } v \text{ is the relationship node } k \end{cases}$$

模型使用门控更新机制重复 T_r 来更新每个节点的隐藏状态， 得到最后每个结点的隐层状态 $\{h_{o_i}^{T_r}, h_{o_j}^{T_r}, h_1^{T_r}, \ldots, h_K^{T_r}\}$，根据初始状态和最终的隐状态计算每个节点的输出特征，聚合这些特征来推断关系

$$f_v^o = o_r([h_v^{T_r}, h_v^0])$$

$$x_{i \to j} = \phi_r([f_{o_i}^o, f_{o_j}^o, f_1^o, \ldots, f_K^o])$$

实验

Comparison with state-of-the-art methods

	Method	SGGen		SGCls		PredCls		
		mR@50	mR@100	mR@50	mR@100	mR@50	mR@100	Mean
Constraint	IMP [30]	0.6	0.9	3.1	3.8	6.1	8.0	3.8
	IMP+ [30, 33]	3.8	4.8	5.8	6.0	9.8	10.5	6.8
	FREQ [33]	4.3	5.6	6.8	7.8	13.3	15.8	8.9
	SMN [33]	5.3	6.1	7.1	7.6	13.3	14.4	9.0
	Ours	**6.4**	**7.3**	**9.4**	**10.0**	**17.7**	**19.2**	**11.7**
Unconstraint	AE [23]	1.6	2.5	6.0	7.8	15.1	19.5	8.8
	IMP+ [30, 33]	5.4	8.0	12.1	16.9	20.3	28.9	15.3
	FREQ [33]	5.9	8.9	13.5	19.6	24.8	37.3	18.3
	SMN [33]	9.3	12.9	15.4	20.6	27.5	37.9	20.6
	Ours	**11.7**	**16.0**	**19.8**	**26.2**	**36.3**	**49.0**	**26.5**

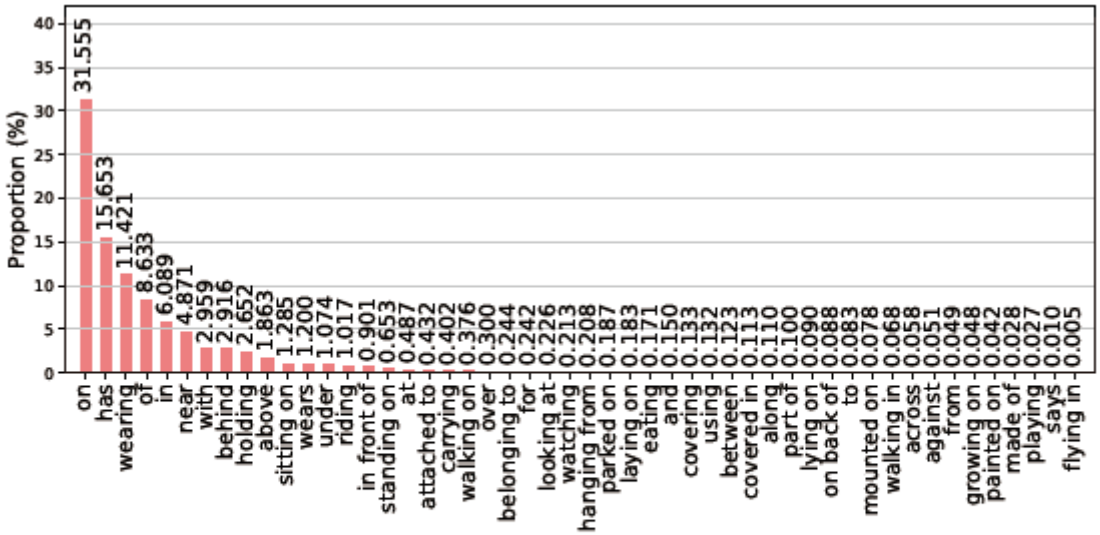
Table 1. Comparison of the mR@50 and mR@100 in % with and without constraint on the three tasks of the VG dataset. We compute Mean mR by averaging mR@50 and mR@100 over the three tasks. As existing works do not present the mR@K metric, we utilize the released models (IMP, FREQ, SMN, AE) or train the model using the released code (IMP+) to generate the results to compute the metric.

在有约束和无约束的情况下，本文方法的mR均值分别为11.7%和26.5%，与之前的最好的方法(即SMN)相比，有30.0%和28.6%的相对提升。

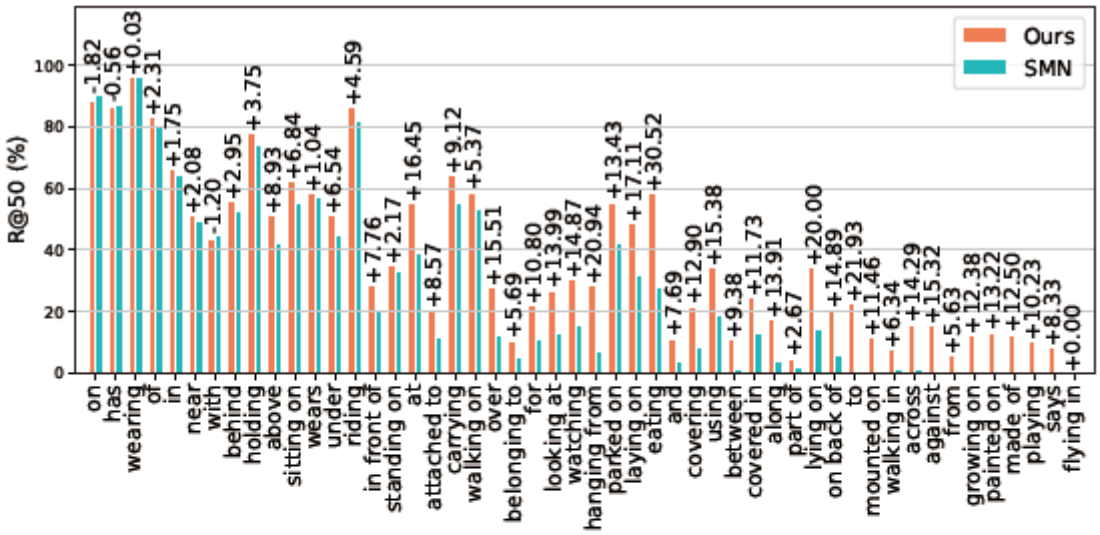
	Methods	SGGen		SGCls		PredCls		
		R@50	R@100	R@50	R@100	R@50	R@100	Mean
Constraint	VRD [19]	0.3	0.5	11.8	14.1	27.9	35.0	14.9
	IMP [30]	3.4	4.2	21.7	24.4	44.8	53.0	25.3
	IMP+ [30, 33]	20.7	24.5	34.6	35.4	59.3	61.3	39.3
	FREQ [33]	23.5	27.6	32.4	34.0	59.9	64.1	40.3
	SMN [33]	**27.2**	**30.3**	35.8	36.5	65.2	67.1	43.7
	Ours	27.1	29.8	**36.7**	**37.4**	**65.8**	**67.6**	**44.1**
No constraint	AE [23]	9.7	11.3	26.5	30.0	68.0	75.2	36.8
	IMP+ [30, 33]	22.0	27.4	43.4	47.2	75.2	83.6	49.8
	FREQ [33]	25.3	30.9	40.5	43.7	71.3	81.2	48.8
	SMN [33]	30.5	35.8	44.5	47.7	81.1	88.3	54.7
	Ours	**30.9**	**35.8**	**45.9**	**49.0**	**81.9**	**88.9**	**55.4**

Table 2. Comparison of the R@50 and R@100 in % with and without constraint on the three tasks of the VG dataset. We compute Mean R by averaging R@50 and R@100 over the three tasks.

在有约束和无约束的情况下，本文方法的平均R分别为44.1%和55.4%，比SMN分别提高了0.4%和0.7%。



(a)



(b)

Figure 4. (a) The distribution of different relationships on the VG dataset. The training and test splits share similar distribution. (b) The R@50 without contraint of our method and the SMN on the predicate classification task on the VG dataset.

我们发现mR@K度量的改进要比R@K度量的改进明显得多。不同关系类型的样本分布是非常不均衡的，前10位出现次数最多的关系占了近90%的样本。与现有方法不同，本文的模型融合先验知识，对语义空间进行显式正则化，因此对于那些不太常见的关系也表现得很好。这样模型就可以很好地解决关系分布不均匀的问题。

Ablative study

Methods	SGGen		SGCls		PredCls		
	mR@50	mR@100	mR@50	mR@100	mR@50	mR@100	Mean
Ours w/o rk & w/o ok	5.1	5.8	6.1	6.5	10.5	11.5	7.6
Ours w/o rk	5.2	5.9	6.5	6.9	11.1	12.0	7.9
Ours	**6.4**	**7.3**	**9.4**	**10.0**	**17.7**	**19.2**	**11.7**
	R@50	R@100	R@50	R@100	R@50	R@100	Mean
Ours w/o rk & w/o ok	25.2	27.9	33.9	34.8	58.7	61.0	40.3
Ours w/o rk	25.5	28.0	34.3	35.2	59.2	61.5	40.6
Ours	**27.1**	**29.8**	**36.7**	**37.4**	**65.8**	**67.6**	**44.1**

Table 3. Comparison of the mR@50, mR@100 (above) and the R@50, R@100 (below) with constraint in % of our full model, our model without relationship correlation (w/o rk), and our model without relationship correlation and object correlation (w/o rk & ok). We compute Mean mR by averaging mR@50 and mR@100 over the three tasks and mean R in the same way.