Representation Learning for Scene Graph Completion via Jointly Structural and Visual Embedding

Hai Wan[1,2,3,*]**, Yonghao Luo**[1]**, Bo Peng**[1]**,** and **Wei-Shi Zheng**[1,3]

[1] School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
[2] Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou, China
[3] Key Laboratory of Machine Intelligence and Advanced Computing (Sun Yat-sen University),
Ministry of Education, China
wanhai@mail.sysu.edu.cn

Abstract

This paper focuses on scene graph completion which aims at predicting new relations between two entities utilizing existing scene graphs and images. By comparing with the well-known knowledge graph, we first identify that each scene graph is associated with an image and each entity of a visual triple in a scene graph is composed of its entity type with attributes and grounded with a bounding box in its corresponding image. We then propose an end-to-end model named Representation Learning via Jointly Structural and Visual Embedding (RLSV) to take advantages of structural and visual information in scene graphs. In RLSV model, we provide a fully-convolutional module to extract the visual embeddings of a visual triple and apply hierarchical projection to combine the structural and visual embeddings of a visual triple. In experiments, we evaluate our model in two scene graph completion tasks: link prediction and visual triple classification, and further analyze by case studies. Experimental results demonstrate that our model outperforms all baselines in both tasks, which justifies the significance of combining structural and visual information for scene graph completion.

1 Introduction

Scene graph, introduced by [Johnson *et al.*, 2015], is a graph-based structural representation which describes the semantic contents of an image. A scene graph is a set of *visual triples* in the form of $(head\ entity, relation, tail\ entity)$ in which an *entity* is composed of its *entity type* with *attributes* and grounded with a bounding box in its corresponding image, and a *relation* is the edge from the *head entity* to the *tail entity*.

The value of scene graph has been confirmed by various Computer Vision and Artificial Intelligence applications. Johnson *et al.* [2015] proposed a conditional random field model of scene graph to retrieve images. Based on the
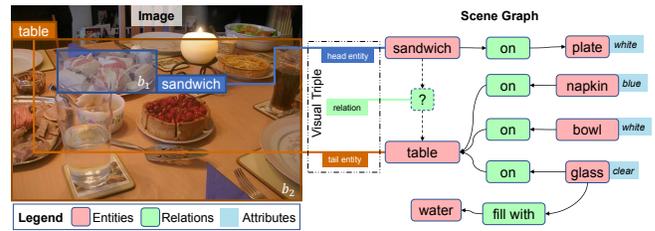

Figure 1: An example image and part of its scene graph from the Visual Genome dataset [Krishna et al., 2017]. The annotation of relation between "*sandwich*" and "*table*" is missing. There are many incomplete visual triples through out the dataset.

scene graph generated from natural language, Anderson *et al.* [2016] presented a method of semantic propositional image caption evaluation. Zhu *et al.* [2017] made use of the entity embeddings of scene graph to enhance visual question answering. Elhoseiny *et al.* [2017] generated facts from scene graph for bidirectional image-fact retrieval. Marino *et al.* [2017] proposed a graph search model treating scene graph as prior knowledge to enhance image classification.

Generating scene graph only from images for the aforementioned tasks has attracted increasing attention. Krishna *et al.* [2017] collected scene graph by crowdsourcing. There are also many automated models that depend on object detectors and the ways of visual relation detection to predict relations. VRD [Lu *et al.*, 2016] considers both visual features of images and language priors of word embeddings in detecting visual triples. SGG [Xu *et al.*, 2017] iteratively generates the entire scene graphs by message passing between the entities and relations in scene graphs. VTransE [Zhang *et al.*, 2017a] learns relation translation vectors from visual triples but without entity embeddings. PPR-FCN [Zhang *et al.*, 2017b] adopts a parallel pairwise and fully-connected structure to enhance performance in detecting visual triples. Other typical automated models include deep relational networks based DR-Net [Dai *et al.*, 2017], reinforcement learning based VRL [Liang *et al.*, 2017], multi-task SGG [Li *et al.*, 2017], VRD with linguistic knowledge distillation [Yu *et al.*, 2017], natural language guided VRD [Liao *et al.*, 2017], *etc.*

However, whether generating scene graph by crowdsourcing or automated models, detecting relations between all

---

*Corresponding author

the entities completely in an image is a time-consuming and difficult job, which makes a scene graph usually suffer from incompletion. As one example in Visual Genome dataset [Krishna *et al.*, 2017] shows in Figure 1, the relation between "*sandwich*" and "*table*" is missing. Intuitively, the relation should be *on* which we can infer from similar visual triples such as $(bowl, on, table)$ of this scene graph or others. Therefore, the issue of *scene graph completion* is addressed here, which aims at predicting new relations between two entities utilizing existing scene graphs and images.

Compared with scene graph, the well-known *knowledge graph* is represented as multi-relational data with enormous fact *triples* in the form of $(head\ entity\ type, relation, tail\ entity\ type)$[1], abridged as (h_t, r, t_t). Knowledge graph *representation learning* is to embed triples into low-dimensional vector spaces, translation-based models of which have been proven effective. TransE [Bordes *et al.*, 2013] considers a relation as translation between a head entity and a tail entity. TransH [Wang *et al.*, 2014] uses a relation-specific hyperplane to project an entity. TransR [Lin *et al.*, 2015] projects an entity using a relation-specific projection matrix. TransD [Ji *et al.*, 2015] constructs a dynamic projection matrix considering both a relation and an entity. Other typical translation-based models include TransG [Xiao *et al.*, 2016], TranSparse [Ji *et al.*, 2016], KG2E [He *et al.*, 2015], *etc.*

Translation-based models in knowledge graphs are good at learning embeddings while preserving the structural information of the graph. However, we cannot simply apply them to scene graphs, since there are three challenges: (1) triples in a knowledge graph are facts that hold in real world, while visual triples in a scene graph are related to the corresponding image; (2) different knowledge graphs can be aggregated if they share the same entity type, while different scene graphs are scarcely possible to be aggregated since the same entity type refers to multiple instances in different images; (3) both structural information of scene graphs and visual information of images should be considered jointly.

To adapt the characteristics of scene graph, we propose a novel end-to-end model named Representation Learning via Jointly Structural and Visual Embedding (RLSV) to take advantages of structural and visual information. Firstly, for a visual triple, we create a fully-convolutional visual feature extraction module to capture the features of entities and relations. Afterwards, we apply three-layered hierarchical projection to combine the structural and visual embeddings of a visual triple. In this process, the visual triple is projected onto attribute space, relation space, and visual space in order, which makes the head entity and the tail entity be packed with attributes, projected onto the same space of the relation, instantiated, and translated by the relation vector.

In experiments, we evaluate our model in two scene graph completion tasks: link prediction and visual triple classification. Experimental results demonstrate that our model performs the best results among all the baselines in both tasks, which justifies the significance of combining structural and visual information in representation learning of scene graph.

[1]Throughout this paper, we identify that the *entity* in knowledge graph is the *entity type* in scene graph.

We further analyze the model capability by case studies.

2 Preliminary

In this section, we give the definition of scene graph and recall two basic translation-based models in knowledge graph.

2.1 Scene Graph

We assume that all images are in a finite set $\mathcal{I}$, and the *entity types* (*resp.*, *relations* and *attributes*) that occur in all scene graphs of $\mathcal{I}$ are in a finite set $\mathcal{E}_t$ (*resp.*, $\mathcal{R}$ and $\mathcal{A}$). We then identify the *visual triples* of scene graph in Definition 1.

Definition 1 (Scene Graph). Given an image $I \in \mathcal{I}$, its scene graph is a set of visual triples $\mathcal{T}_I \subseteq \mathcal{E}_I \times \mathcal{R}_I \times \mathcal{E}_I$, where $\mathcal{E}_I$ is the *entity* set, $\mathcal{R}_I$ is the relation set, and $\mathcal{R}_I \subseteq \mathcal{R}$. Each entity $e_{I,k} = (e_{t,I,k}, \mathcal{A}_{I,k}, b_{I,k}) \in \mathcal{E}_I$ has an entity type $e_{t,I,k} \in \mathcal{E}_t$, where $k \in \{1, \ldots, |\mathcal{E}_I|\}$. Entity $e_{I,k}$ is packed with attributes $\mathcal{A}_{I,k} \subseteq \mathcal{A}$ and grounded with a bounding box $b_{I,k}$ in image I. A visual triple is of the form $(h, r, t) \in \mathcal{T}_I$, where the *head* entity and the *tail* entity $h, t \in \mathcal{E}_I$ and the relation $r \in \mathcal{R}_I$.

There are 5 visual triples in the scene graph $\mathcal{T}_I$ of Figure 1, *e.g.*, $((sandwich, \{\}, b_1), on, (plate, \{white\}, b_2))$. For simplicity, we write as $(sandwich, on, plate)$.

Intuitively, Definition 1 characterizes that: (1) each scene graph is associated with an image; (2) each entity in a scene graph is composed of its entity type with attributes and grounded with a bounding box in its corresponding image, while each entity in knowledge graph is an entity type.

Scene graph completion is to find more visual triples for a scene graph utilizing existing scene graphs and images.

Definition 2 (Scene Graph Completion). Given an image I and its scene graph $\mathcal{T}_I$, $\exists e_{I,p}, e_{I,q} \in \mathcal{E}_I$ and $\nexists r_I \in \mathcal{R}_I$ s.t. $(e_{I,p}, r_I, e_{I,q}) \in \mathcal{T}_I$, where $p \neq q$. Scene graph completion is to find new visual triples $(e_{I,p}, r_I^+, e_{I,q})$ s.t. $r_I^+ \in \mathcal{R}$, and then extend $\mathcal{T}_I$ to be $\mathcal{T}_I^+ = \mathcal{T}_I \cup \{(e_{I,p}, r_I^+, e_{I,q}) | e_{I,p}, e_{I,q} \in \mathcal{E}_I, r_I^+ \in \mathcal{R}, p \neq q\}$ s.t. $|\mathcal{T}_I^+| > |\mathcal{T}_I|$.

As shown in Figure 1, we can add a new visual triple $(sandwich, on, table)$ to $\mathcal{T}_I$ s.t. $|\mathcal{T}_I^+| = 6 > |\mathcal{T}_I| = 5$.

2.2 Knowledge Graph Embedding

TransE [Bordes *et al.*, 2013] is the first translation-based model and TransD [Lin *et al.*, 2015] extends TransE by considering the diversity of both relations and entities.

TransE. For each triple (h_t, r, t_t) in a knowledge graph, TransE wants $\boldsymbol{h}_t + \boldsymbol{r} \approx \boldsymbol{t}_t$ when (h_t, r, t_t) holds ($\boldsymbol{h}_t, \boldsymbol{r}, \boldsymbol{t}_t$ denote the column vectors of a triple (h_t, r, t_t) respectively). It indicates that $\boldsymbol{t}_t$ should be the nearest entity from $(\boldsymbol{h}_t + \boldsymbol{r})$. Thus, TransE defines the following scoring function:

$$E(h_t, r, t_t) = \|\boldsymbol{h}_t + \boldsymbol{r} - \boldsymbol{t}_t\|_{L_1/L_2}. \tag{1}$$

The function returns low score if (h_t, r, t_t) holds, vice versa.

TransD. For each triple (h_t, r, t_t), TransD assigns h_t, t_t, and r with extra projection vectors $\boldsymbol{h}_{tp}$, $\boldsymbol{t}_{tp}$, and $\boldsymbol{r}_p$. They set two mapping matrices $\mathbf{M}_{rh_t}$, $\mathbf{M}_{rt_t} \in \mathbb{R}^{m \times n}$ to project entities from entity space to relation space as follows:

$$\mathbf{M}_{rh_t} = \boldsymbol{r}_p \boldsymbol{h}_{tp}^{\mathrm{T}} + \mathbf{I}^{m \times n}, \qquad \mathbf{M}_{rt_t} = \boldsymbol{r}_p \boldsymbol{t}_{tp}^{\mathrm{T}} + \mathbf{I}^{m \times n}. \tag{2}$$

$\mathbf{M}_{rh_t}$ and $\mathbf{M}_{rt_t}$ are determined by both entities and relations, which makes the two projection vectors interact sufficiently.
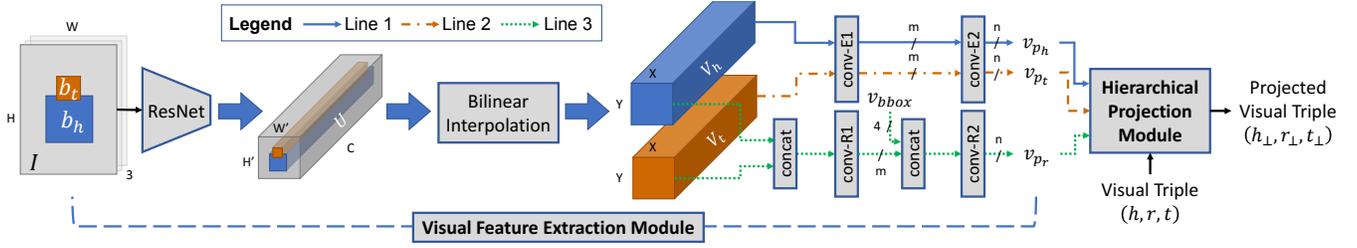
Figure 2: Architecture of RLSV model.

3 Methodology

To jointly embed the structural and visual information of scene graph, we construct a model named RLSV. We assume that each embedding takes value in $\mathbb{R}^n$.

3.1 RLSV Architecture

For a visual triple (h, r, t), we propose a *visual feature extraction module* and a *hierarchical projection module* to jointly combine structural embeddings (h, r, t) and visual embeddings $(v_{p_h}, v_{p_r}, v_{p_t})$ as new representations $(h_\perp, r_\perp, t_\perp)$. Following Equation 1 of TransE, we define the scoring function of a visual triple (h, r, t) of scene graph $\mathcal{T}_I$ as follow:

$$E_I(h, r, t) = \|h_\perp + r_\perp - t_\perp\|_{L_1/L_2}. \quad (3)$$

As illustrated in Figure 2, RLSV model is an end-to-end architecture that accepts an image I and a visual triple (h, r, t) as inputs, and measures their score by Equation 3. The visual feature extraction model is to embed the inputting image as visual projection vectors which are used later for constructing projection matrices. The hierarchical projection module is to project a visual triple onto attribute space, relation space, and visual space in order. It makes the head entity and the tail entity be packed with attributes, projected onto the same space of the relation, instantiated, and translated by the relation vector.

3.2 Visual Feature Extraction

The visual feature extraction module aims at retaining the visual information of a visual triple. We encode an image I of shape $H \times W \times 3$ by ResNet-50 [He *et al.*, 2016] which consists of 5 fully-convolutional blocks. We extract the output of the fourth block as a global feature map $\mathbf{U}$ of shape $H' \times W' \times C$, where $H' = \lfloor \frac{W}{32} \rfloor$, $W' = \lfloor \frac{H}{32} \rfloor$, and $C = 1024$.

Obviously, the bounding boxes of the head entities and the tail entities are in various sizes even though they are projected onto $\mathbf{U}$. Thus, we adopt bilinear interpolation [Jaderberg *et al.*, 2015; Johnson *et al.*, 2016; Zhang *et al.*, 2017a] to extract the local features of the entities for its smoothness in gradient backpropagation. Specifically, given a feature map $\mathbf{U}$ and an projected entity bounding box on $\mathbf{U}$, we can get a local feature map of entity $\mathbf{V}$ in convolutional style as follow:

$$V_{i,j,c} = \sum_{i'}^{H'} \sum_{j'}^{W'} U_{i',j',c} k(i' - G_{i,j,1}) k(j' - G_{i,j,2}), \quad (4)$$

where $\mathbf{G}$ is a sampling grid of shape $Y \times X \times 2$ marking real-valued coordinates of $\mathbf{V}$ on $\mathbf{U}$ and $k(d) = \max(0, 1 - |d|)$ is

the bilinear interpolation kernel. By bilinear interpolation, the bounding box region of the head b_h and that of the tail entity b_t are extracted as local feature maps $\mathbf{V}_h$ and $\mathbf{V}_t$ respectively.

For the head entity h and the tail entity t, they share the same embedding pipeline (Line 1 and 2 in Figure 2) composed of 1×1 convolutional layers *conv-E1* and *conv-E2*. There is a global averaging pooling layer [Lin *et al.*, 2014] between *conv-E1* and *conv-E2*. Thus, $\mathbf{V}_h$ (*resp.*, $\mathbf{V}_t$) is then embedded as the visual projection vector v_{p_h} (*resp.*, v_{p_t}).

For the relation r, the embedding pipeline is similarly composed of 1×1 convolutional layers *conv-R1* and *conv-R2* (Line 3 in Figure 2). But for layer *conv-R1*, we concatenate $\mathbf{V}_h$ and $\mathbf{V}_t$ along the channel axis as the visual information. Before embedding with layer *conv-R2*, we also concatenate bounding box feature v_{bbox} as the location information, where v_{bbox} is a 4-dimensional vector (t_x, t_y, t_w, t_h) which is the scale-invariant parameterization of two bounding boxes [Girshick, 2015]. We assume that the bounding box of the head (*resp.*, tail) entity is (x, y, w, h) (*resp.*, (x', y', w', h')), and the formulation of the scalars of v_{bbox} are as follows:

$$t_x = \frac{x - x'}{w'}, \ t_y = \frac{y - y'}{h'}, \ t_w = \log \frac{w}{w'}, \ t_h = \log \frac{h}{h'}, \ (5)$$

where x, y are the left-top coordinates of the bounding box and w, h are its width and height (likewise for $x', y', w',$ and h'). The location information is to better characterize spatial features between two entities, which is useful in describing spatial relations (*e.g.*, *on top of*) and some verbal relations (*e.g.*, *cover*). And the visual information can distinguish different relations with similar location information (*e.g.*, *ride* and *sit*). By combining the location and visual information, we get the visual projection vector v_{p_r} for the relation r.

3.3 Hierarchical Projection

We devise a three-layered hierarchical projection module to combine the structural embeddings and the visual embeddings of a visual triple. Given the structural embeddings (h, r, t) and the visual embeddings $(v_{p_h}, v_{p_r}, v_{p_t})$ of a visual triple (h, r, t), we construct three dynamic projection matrices $\mathbf{M}_e^a, \mathbf{M}_e^r, \mathbf{M}_e^v \in \mathbb{R}^{n \times n}$ (e can be h or t) to project the head entity and the tail entity onto attribute space, relation space, and visual space in order as follows:

$$h_\perp = \mathbf{M}_h^v \mathbf{M}_h^r \mathbf{M}_h^a h, \qquad t_\perp = \mathbf{M}_t^v \mathbf{M}_t^r \mathbf{M}_t^a t. \quad (6)$$

Meanwhile, the relation is projected onto visual space via dynamic projection matrix $\mathbf{M}_r^v \in \mathbb{R}^{n \times n}$ as follow:

$$r_\perp = \mathbf{M}_r^v r. \quad (7)$$

Finally, we use Equation 3 to score $(h_\perp, r_\perp, t_\perp)$.

To construct dynamic projection matrices, referring to the method of TransD in Equation 2, we assign each entity type (*resp.*, relation and attribute) with a projection vector $\boldsymbol{e}_p \in \mathbb{R}^n$ (*resp.*, $\boldsymbol{r}_p$ and $\boldsymbol{a}_p$), where p denotes the projection vector.

Attribute Space.   We assume that the entities are originally in entity space and an entity projected onto attribute space is to be packed with its attributes. Given an entity e and its attribute set $\mathcal{A}_e$, we set the number of attributes $N_a = |\mathcal{A}_e|$. For each $a_i \in \mathcal{A}_e$, we construct the attribute projection matrix as follow:

$$\mathbf{M}_e^{a_i} = \boldsymbol{a}_{ip}\boldsymbol{e}_p^{\mathrm{T}} + \mathbf{I}^{n \times n}, \tag{8}$$

where $\mathbf{I}^{n \times n}$ is an identity matrix. However, an entity may have multiple attributes, we have to merge these attribute projection matrices. A simple method is to use weighted summation to blend multiple attribute projection matrices as follow:

$$\mathbf{M}_e^{a} = \sum_{i=1}^{N_a} \beta_i \mathbf{M}_e^{a_i} = \left(\sum_{i=1}^{N_a} \beta_i \boldsymbol{a}_{ip}\right) \boldsymbol{e}_p^{\mathrm{T}} + \mathbf{I}^{n \times n}, \tag{9}$$

where β_i denotes the weight of each attribute a_i and $\sum_{i=1}^{N_a} \beta_i = 1$. Considering the attributes may have different contributions with respect to the visual information of the corresponding entity, the attention mechanism is introduced to compute the weights based on the entity visual embedding $\boldsymbol{v}_{p_e}$ and the attribute projection vector $\boldsymbol{a}_{ip}$ as follows:

$$\varepsilon_i = \boldsymbol{w}_b^T \tanh(\mathbf{W}_v \boldsymbol{v}_{p_e} + \mathbf{W}_a \boldsymbol{a}_{ip}), \tag{10}$$

$$\beta_i = \mathrm{softmax}(\varepsilon_i) = \frac{\exp(\varepsilon_i)}{\sum_{j=1}^{N_a} \exp(\varepsilon_j)}, \tag{11}$$

where $\boldsymbol{w}_b \in \mathbb{R}^n$ and $\mathbf{W}_v, \mathbf{W}_a \in \mathbb{R}^{n \times n}$ are learnable weights. Particularly, for an entity without any attribute, *i.e.*, $N_a = 0$, we set $\mathbf{M}_e^a = \mathbf{I}^{n \times n}$, which means there is no need to project onto attribute space. From Equation 8-11, we get the projection matrix $\mathbf{M}_h^a$ (*resp.*, $\mathbf{M}_t^a$) for the head (*resp.*, tail) entity.

Relation Space.   Although the dimension of entities and relations may be the same, they are not in the same sematic space since they are different types of vectors. We also discover that the vast majority of relations are many-to-many, so we have to project each entity from attribute space to relation space. Thus, we construct the visual projection matrix for the head (*resp.*, tail) entity by its visual embedding $\boldsymbol{v}_{p_h}$ (*resp.*, $\boldsymbol{v}_{p_t}$) and the relation projection vector $\boldsymbol{r}_p$ as follows:

$$\mathbf{M}_h^r = \boldsymbol{r}_p \boldsymbol{v}_{p_h}^{\mathrm{T}} + \mathbf{I}^{n \times n}, \qquad \mathbf{M}_t^r = \boldsymbol{r}_p \boldsymbol{v}_{p_t}^{\mathrm{T}} + \mathbf{I}^{n \times n}. \tag{12}$$

Visual Space.   The entities and the relation in a visual triple is not immutable, *e.g.*, $(person, stand\ on, ground)$ does not always hold. Therefore, a visual triple must be incorporated with a particular image to evaluate its correctness. We assign the head entity, the tail entity and the relation in a visual triple with a projection matrix respectively and project them onto visual space, with which a visual triple is instantiated in visual space. We construct the visual projection matrices by their visual embeddings and projection vectors as follows:

$$\begin{aligned} \mathbf{M}_h^v &= \boldsymbol{v}_{p_h} \boldsymbol{h}_p^{\mathrm{T}} + \mathbf{I}^{n \times n}, \\ \mathbf{M}_t^v &= \boldsymbol{v}_{p_t} \boldsymbol{t}_p^{\mathrm{T}} + \mathbf{I}^{n \times n}, \\ \mathbf{M}_r^v &= \boldsymbol{v}_{p_r} \boldsymbol{r}_p^{\mathrm{T}} + \mathbf{I}^{n \times n}. \end{aligned} \tag{13}$$

3.4 Objective

We formalize a max-margin function with negative sampling as the training objective:

$$L = \sum_{I \in \mathcal{I}} \sum_{(h,r,t) \in \mathcal{T}_I} \sum_{(h',r',t') \in \mathcal{T}_I'} [E_I(h,r,t) + \gamma - E_I(h',r',t')]_+, \tag{14}$$

where $[x]_+ \triangleq \max(0, x)$ and γ is a margin hyperparameter. $\mathcal{T}_I'$ stands for the negative sampled visual triple set generated from positive visual triple set $\mathcal{T}_I$. We define $\mathcal{T}_I'$ as follow:

$$\begin{aligned} \mathcal{T}_I' = \{(h',r,t)|h' \in \mathcal{E}_I\} &\cup \{(h,r,t')|t' \in \mathcal{E}_I\} \\ &\cup \{(h,r',t)|r' \in \mathcal{R}\}, \quad (h,r,t) \in \mathcal{T}_I. \end{aligned} \tag{15}$$

The entities and relation in a visual triple can be randomly replaced by another one, but we assure $\mathcal{T}_I \cap \mathcal{T}_I' = \emptyset$. We constrain the norms of both original and projected embeddings of a visual triple as $\|\boldsymbol{h}\|_2 \leqslant 1$, $\|\boldsymbol{t}\|_2 \leqslant 1$, $\|\boldsymbol{r}\|_2 \leqslant 1$, $\|\boldsymbol{h}_\perp\|_2 \leqslant 1$, $\|\boldsymbol{t}_\perp\|_2 \leqslant 1$, and $\|\boldsymbol{r}_\perp\|_2 \leqslant 1$.

3.5 Optimization and Implementation Details

The learnable parameter set of RLSV model can be formalized as $\theta = (\mathbf{E}, \mathbf{R}, \mathbf{E}_p, \mathbf{R}_p, \mathbf{A}_p, \mathbf{W})$, where $\mathbf{E}$ (*resp.*, $\mathbf{R}$) represents the entity (*resp.*, relation) embeddings; $\mathbf{E}_p$, $\mathbf{R}_p$, and $\mathbf{A}_p$ represent the projection vector sets of entity, relation, and attribute; and $\mathbf{W}$ stands for the weights of each neural layer.

For model initialization, we pre-train ResNet-50 on the entities in the scene graph datasets and randomly initialize the rest of neural layers as in [He *et al.*, 2016]. $\mathbf{E}$, $\mathbf{R}$, $\mathbf{E}_p$, $\mathbf{R}_p$, and $\mathbf{A}_p$ are also randomly initialized by Xavier initializer [Glorot and Bengio, 2010]. We implement Bernoulli strategy [Wang *et al.*, 2014] for generating negative head or tail entities with different probabilities. For efficiency, we freeze the ResNet CNN and train the rest of RLSV by Adam [Kingma and Ba, 2015]. We implement on Tensorflow. One mini-batch of 20 visual triples runs in approximately 70ms on a GTX 1080 Ti GPU with 8 or 9 epoches for the model to converge.

4 Experiments

We evaluate our model with two typical scene graph completion tasks[2]: link prediction and visual triple classification. We first introduce datasets with experimental settings, and then analyze the experimental results in detail.

4.1 Datasets

Our experiments used two large-scale scene graph datasets:

- **VRD**: Visual Relationship Dataset [Lu *et al.*, 2016] is built upon the raw annotation of Scene Graph dataset [Johnson *et al.*, 2015], containing 5,000 images with 100 entity types and 70 relations. We keep top-100 attributes and link each entity with their attributes using the raw annotation, resulting in 37,339 visual triples in total.

---

[2]All experiments run in 64-bit Linux Ubuntu 16.04.4 LTS on a machine with a 2.20 GHz Intel Xeon E5-2630 CPU, a GeForce GTX 1080 Ti GPU, and a 128G 2133 MHz memory. For more information about codes and data, please visit `https://github.com/sysulic/RLSV`.

Figure 3 examples

VRD

#1 — umbrella (black, wet) → cover → person (walking, white, caucasian)

	RANK	Ans#1	Ans#2	Ans#3	Ans#4	Ans#5
RLSV-H	7	above	on	over	behind	next to
RLSV-V+H	1	cover	next to	above	near	behind

#2 — person → ride → motorcycle

	RANK	Ans#1	Ans#2	Ans#3	Ans#4	Ans#5
RLSV-H	9	on	next to	in the front of	beside	behind
RLSV-V+H	2	on	ride	next to	in the front of	beside

#3 — dog (white) → in → basket (white)

	RANK	Ans#1	Ans#2	Ans#3	Ans#4	Ans#5
RLSV-H	13	next to	near	in the front of	hold	on
RLSV-V+H	5	sit on	next to	near	in the front of	in

VG

#4 — people → cut → cake

	RANK	Ans#1	Ans#2	Ans#3	Ans#4	Ans#5
RLSV-H	21	on	have	hold	with	near
RLSV-V+H	5	on	behind	look at	next to	cut

#5 — wall → support → building

	RANK	Ans#1	Ans#2	Ans#3	Ans#4	Ans#5
RLSV-H	20	on	of	on side of	in	have
RLSV-V+H	19	on	of	in front of	next to	on side of

#6 — boy → wear → pants (blue)

	RANK	Ans#1	Ans#2	Ans#3	Ans#4	Ans#5
RLSV-H	1	wear	have	in	with	on
RLSV-V+H	1	wear	in	have	with	on

Figure 3: Qualitative examples of relation prediction. On the left side, we show an image with a testing visual triple from VRD and VG. One the right side, we show the correct relation rankings (denoted as "RANK") and top-5 answers (denoted as "Ans#") from RLSV-H and RLSV-V+H. Relations in bold, italic, and underline fonts denote the correct, plausible, and wrong answers respectively.

Dataset	$|\mathcal{E}_t|$	$|\mathcal{R}|$	$|\mathcal{A}|$	#Train	#Valid	#Test
VRD	100	70	100	26,057	5,641	5,641
VG	200	100	100	657,404	141,627	141,627

Table 1: Statistics of datasets

Element	Relation		Head		Tail	
	raw	filt	raw	filt	raw	filt
VRD	70.00	69.77	8.56	8.27	8.56	8.11
VG	100.00	99.90	16.81	15.97	16.81	16.07

Table 2: Averaged number of the replaceable relations, head and tail entities per visual triple on VRD and VG.

- VG: The latest version Visual Genome relation and attribute dataset. We used the entity alias and the relation alias provided by [Krishna *et al.*, 2017]. Following the experiments in [Zhang *et al.*, 2017a], we keep 99,993 images with 200 entity types, 100 relations, and 100 attributes, resulting in 940,658 visual triples in total.

To fulfill the completion setting of RLSV model[3], we split approximately 70% of visual triples per image for training and the rest for validation and testing as shown in Table 1.

4.2 Experimental Settings

We set the dimension of embeddings $n = 200$ and the number of channels $m = 1024$ for *conv-E1* and *conv-R1*. The number

---

[3]Compared with our experimental setting, the setting of generating scene graph only from images is that, a dataset is split by images, *e.g.*, 4000 images for training and 1000 testing on VRD dataset.

---

of attributes per entity N_a is up to 6. In Equation 3, we use L_1 norm to measure vector distance. In Equation 14, we select margin $\gamma = 1.0$. The initial learning rate is 1×10^{-3} and descends through iterations until 6.25×10^{-5}.

We compared the following methods in the experiments:

- RLSV-V+H: Our model introduced in Section 3.

- RLSV-V: Part of our model. We only use visual feature extraction module to embed relation and connect *conv-R2* layer with softmax function as a relation classifier.

- RLSV-H: Part of our model, using only hierarchical projection module without projecting onto visual space. Since there is no visual embedding, we average the attribute projection vectors by setting $\beta_i = \frac{1}{N_a}$ and replace $\boldsymbol{v}_{p_h}, \boldsymbol{v}_{p_t}$ by $\boldsymbol{h}_p, \boldsymbol{t}_p$ in Equation 12 respectively.

- Generating scene graph only from images: VTransE and PPR-FCN, which are modified for our settings.

- Translation-based models: TransE, TransH, TransR and TransD, aggregating all scene graphs.

- Rand: Random permutation in ranking the answers.

4.3 Link Prediction

Link prediction is to complete a visual triple (h, r, t) with one of head entity h, relation r, or tail entity t is missing.

Evaluation Protocol. For each testing visual triple of image I, we replaced the relation r by each relation in $\mathcal{R}$. We also replaced the head h or the tail t by each entity in $\mathcal{E}_I$. The score of each replaced visual triple is determined by Equation

Dataset																
	VRD								VG							
Metric	rAVG		Hits@1 (%)		Hits@5 (%)		Hits@10 (%)		rAVG		Hits@1 (%)		Hits@5 (%)		Hits@10 (%)	
	raw	filt	raw	filt	raw	filt	raw	filt	raw	filt	raw	filt	raw	filt	raw	filt
Rand	35.71	35.59	1.63	1.63	6.98	7.02	14.02	14.06	50.56	50.51	1.02	1.02	5.01	5.01	10.01	10.02
TransE	5.34	5.22	28.01	29.39	66.85	67.56	89.45	89.81	3.76	3.71	50.19	50.99	82.92	83.20	91.56	91.73
TransH	5.22	5.11	27.76	29.07	67.06	67.81	89.66	90.29	3,85	3.80	47.39	48.06	82.59	82.95	91.86	92.01
TransR	5.28	5.17	21.54	22.41	73.71	74.51	89.90	90.52	3.74	3.68	42.52	43.24	83.36	83.67	92.89	93.03
TransD	5.22	5.10	29.80	31.04	68.94	69.42	85.37	85.94	3.78	3.73	45.94	46.70	83.53	83.79	92.13	92.27
VTransE	5.17	4.97	38.29	44.09	75.57	76.58	87.32	87.79	4.24	4.19	50.92	52.04	82.82	83.10	90.68	90.80
PPR-FCN	4.96	4.75	34.05	38.77	76.32	77.75	87.70	88.14	4.32	4.28	50.22	51.17	82.11	82.40	90.39	90.51
RLSV-V	7.07	6.96	28.62	29.27	62.44	63.00	80.16	80.62	6.61	6.57	37.29	37.70	73.75	74.02	83.74	83.86
RLSV-H	3.66	3.54	48.69	50.97	82.18	83.09	93.26	93.51	3.13	3.08	57.19	58.43	87.60	87.85	94.06	94.18
RLSV-V+H	**3.59**	**3.46**	**49.32**	**51.68**	**83.23**	**84.15**	**93.33**	**93.62**	**3.03**	**2.98**	**58.37**	**59.66**	**87.83**	**88.10**	**94.52**	**94.64**

Table 3: Results of relation prediction on VRD and VG.

Dataset																
	VRD								VG							
Entity	Head				Tail				Head				Tail			
Metric	rAVG		Hits@5 (%)		rAVG		Hits@5 (%)		rAVG		Hits@5 (%)		rAVG		Hits@5 (%)	
	raw	filt	raw	filt	raw	filt	raw	filt	raw	filt	raw	filt	raw	filt	raw	filt
Rand	4.80	4.65	64.47	66.55	4.82	4.58	64.51	67.70	8.92	8.50	38.08	40.70	8.92	8.55	38.14	40.62
VTransE	2.96	2.76	86.46	88.53	3.09	2.78	85.55	88.10	5.08	4.64	68.62	72.28	4.67	4.30	72.38	75.56
PPR-FCN	2.76	2.60	88.46	90.44	2.77	2.53	88.71	91.28	5.46	5.04	64.68	68.26	5.03	4.66	68.24	71.51
RLSV-V	3.29	3.15	82.61	84.46	3.26	3.04	82.68	85.34	6.94	6.53	54.47	57.35	7.28	6.90	49.11	52.27
RLSV-V+H	**2.42**	**2.26**	**93.32**	**94.42**	**2.53**	**2.30**	**92.32**	**93.94**	**3.89**	**3.47**	**78.09**	**81.89**	**3.56**	**3.19**	**81.26**	**84.60**

Table 4: Results of entity prediction on VRD and VG.

3 and we ranked all scores of replaced visual triples in ascending order. Specially, for RLSV-V, VTransE and PPF-FCN, we used the input of the classifier as score. It is the "raw" settings [Bordes *et al.*, 2013] and we also used the "filt" settings which wipe out the replaced visual triples already existing in the dataset. Table 2 summarizes the averaged number of the replaceable relations, head and tail entities per visual triple.

Following [Bordes *et al.*, 2013], we report the results of two evaluation metrics: (1) the averaged rank of correct predictions in the testing set (rAVG, the lower the better); (2) proportion of a correct prediction that ranks in top-k (Hits@k, where $k = \{1, 5, 10\}$ and the higher the better).

Results of Relation Prediction. From Table 3 and Figure 3, we observe that: (1) the joint model RLSV-V+H significantly outperforms its separate variations RLSV-V, RLSV-H, and other baselines on both evaluation metrics of rAVG and Hits@k, especially on VRD. It indicates that the combination of structural and visual information leads to better performance on relation prediction. As the example $(person, ride, motorcycle)$ in Image #2 shown in Figure 3, RLSV-V+H is able to learn the meaning of $ride$ from image, bringing it to a higher ranking. (2) Although RLSV-H separately models structural information, it still outperforms VTransE and PPR-FCN because negative sampling strengthens the ability of distinguishing wrong answers. (3) Since the translation-based models regard the whole dataset as a large aggregated scene graph, they confuse the entities of the same type rather than the same instance and the visual information of scene graph is lost, resulting in relatively weaker performance. (4) Although there are some relations hard to model, *e.g.*, $(wall, support, building)$ in Image #5 shown in Figure 3, both RLSV-V+H and RLSV-H still give plausible answers.

Results of Entity Prediction. Since the translation-based models and RLSV-H cannot distinguish different entities in images, we only show the results of RLSV-V+H, RLSV-H, VTransE, and PPR-FCN. From the results in Table 4, we have: (1) RLSV-V+H outperforms other baselines on both evaluation metrics of rAVG and Hits@5, which indicates that our model performs better in capturing the relations between correct entity pairs. (2) Although the number of replaceable head and tail entities is much less than relations, RLSV-V+H surpasses VTransE on rAVG by 0.50 on VRD and 1.17 on VG when predicting head entities under the "filt" settings.

4.4 Visual Triple Classification

This task is to confirm whether a given visual triple (h, r, t) is correct or not, *i.e.*, binary classification on a visual triple.

Evaluation Protocol. Following [Socher *et al.*, 2013], we generated the negative visual triples on VRD and VG datasets by randomly replacing relation with another one. We assured that the number of positive and negative visual triples is equal. If the score of a visual triple (h, r, t) is below a relation-specific threshold σ_r, then it is positive, and vise versa. The threshold σ_r is determined on the validation set by maximizing the relation-specific classification accuracies.

Results. Since the translation-based models and RLSV-H cannot distinguish different entities in images, we only show the results of RLSV-V+H, RLSV-H, VTransE, and PPR-FCN. From Table 5, we observe that: (1) RLSV-V+H outperforms RLSV-V and PPR-FCN. By the translation-based objective, RLSV-V+H can represent the similar relations such as $above$ and on, making the classification thresholds more appropriate. While RLSV-V and PPR-FCN fails because it treats each relation as a unique category. (2)

Metric	Accuracy (%)	
Dataset	VRD	VG
VTransE	87.64	91.62
PPR-FCN	86.53	90.23
RLSV-V	86.17	87.58
RLSV-V+H	**90.31**	**93.05**

Table 5: Results of visual triple classification on VRD and VG.

RLSV-V+H also outperforms VTransE, which confirms the improvements introduced by entity embeddings.

5 Case Studies

In this section, we analyze our model in two cases. One is the completion in a full scene graph, and the other is to illustrate the semantic affinities of entity and relation embeddings.

5.1 Full Scene Graph Completion

To demonstrate the ability of RLSV model in completing full scene graphs, we trained our model with the whole dataset and computed the visual triple classification thresholds using the validation set. Our model accepted the top-1 relation in a visual triple if its score is below the relation-specific thresholds. In VRD, we observed that there are 217,197 new visual triples in 4,637 images. In VG, we randomly sampled 5,000 images with 892,112 new visual triples. Figure 4 shows that around 89% of the new visual triples in VRD and VG are accepted by our model. In Figure 1, *e.g.*, our model accepted $(sandwich, on, table)$, $(plate, near, glass)$, and $(plate, near, bowl)$. To evaluate the correctness of the predicted relations of the accepted visual triples, we randomly sampled 200 images and asked human subjects to check their top-5 relations. We observed that over 70% of the accepted visual triples in top-1 on VRD and VG are correct answers, demonstrating the effectiveness and robustness of our model. Thus, there is a typical application scenario of RLSV model, where human is recommended to select one or more correct relations to complete scene graph.

5.2 Semantic Affinities of Embeddings

By t-SNE visualization [Maaten and Hinton, 2008], we discovered semantic affinities among the entity type and relation embeddings of our model. Among the entity type embeddings, there are some apparent clusters such as apparel and accessories (*e.g.*, $shorts$, $pants$, $shoes$), plains (*e.g.*, $road$, $street$, $grass$), and animals (*e.g.*, cat, dog, $bear$). Among the relation embeddings, *e.g.*, $beneath$, $under$, and $below$ seem similar, staying away from their opposites $above$ and $over$.

6 Conclusion and Future Work

By comparing scene graph with knowledge graph, we have identified the visual features of visual triple in scene graph. Then we have proposed a novel RLSV model for scene graph completion. The RLSV model utilizes a visual feature extraction module composed of series of fully-convolutional layers to embed visual information, combines the structural embeddings of visual triples by hierarchical projection module, and
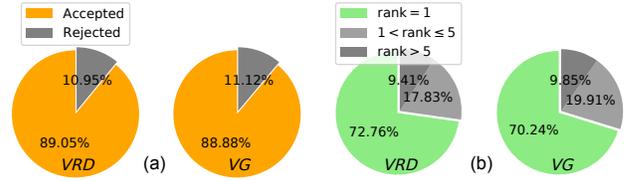
Figure 4: (a) Distribution of new visual triples; (b) Evaluation of human subjects on the relation rankings of the accepted visual triples.
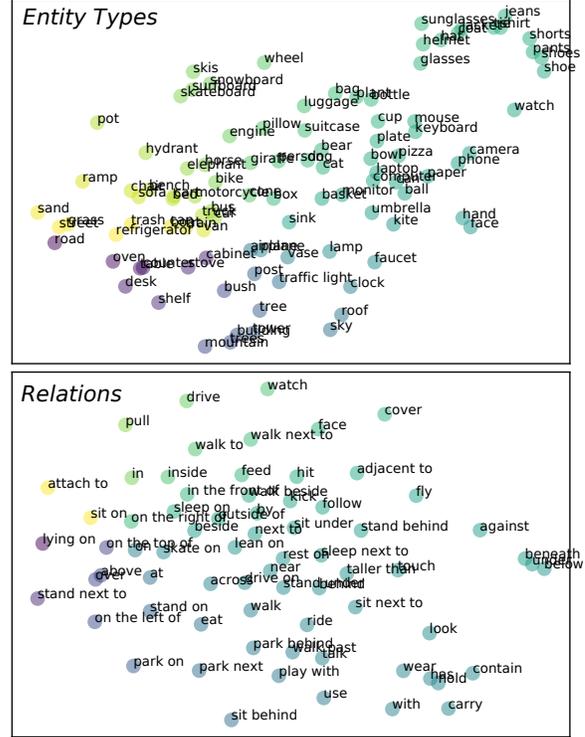
Figure 5: Entity and relation embeddings visualization on VRD.

finally represents entity and relation translation-based models in visual space. Experimental results of link prediction and visual triple classification have clearly demonstrated that our model has a good potential in scene graph completion.

To the best of our knowledge, this is the first attempt to present representation learning for scene graph completion. By studying the differences between knowledge graph and scene graph, we have succeeded in exploiting translation-based models in knowledge graph for scene graph completion. Future work will focus on adapting methods in knowledge graph to scene graph applications, *e.g.*, visual question answering, generating scene graph with text descriptions, *etc.*

Acknowledgments

References

[Anderson *et al.*, 2016] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. SPICE: semantic propositional image caption evaluation. In *Proceedings of ECCV*, pages 382–398, 2016.

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of NIPS*, pages 2787–2795, 2013.

[Dai *et al.*, 2017] Bo Dai, Yuqi Zhang, and Dahua Lin. Detecting visual relationships with deep relational networks. In *Proceedings of CVPR*, pages 3298–3308, 2017.

[Elhoseiny *et al.*, 2017] Mohamed Elhoseiny, Scott Cohen, Walter Chang, Brian L. Price, and Ahmed M. Elgammal. Sherlock: Scalable fact learning in images. In *Proceedings of AAAI*, pages 4016–4024, 2017.

[Girshick, 2015] Ross B. Girshick. Fast R-CNN. In *Proceedings of ICCV*, pages 1440–1448, 2015.

[Glorot and Bengio, 2010] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of AISTATS*, pages 249–256, 2010.

[He *et al.*, 2015] Shizhu He, Kang Liu, Guoliang Ji, and Jun Zhao. Learning to represent knowledge graphs with gaussian embedding. In *Proceedings of CIKM*, pages 623–632, 2015.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of CVPR*, pages 770–778, 2016.

[Jaderberg *et al.*, 2015] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Proceedings of NIPS*, pages 2017–2025, 2015.

[Ji *et al.*, 2015] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of ACL*, pages 687–696, 2015.

[Ji *et al.*, 2016] Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. Knowledge graph completion with adaptive sparse transfer matrix. In *Proceedings of AAAI*, pages 985–991, 2016.

[Johnson *et al.*, 2015] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. Image retrieval using scene graphs. In *Proceedings of CVPR*, pages 3668–3678, 2015.

[Johnson *et al.*, 2016] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Processings of CVPR*, pages 4565–4574, 2016.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Processings of ICLR*, 2015.

[Krishna *et al.*, 2017] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowd-sourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

[Li *et al.*, 2017] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *Proceedings of ICCV*, pages 1270–1279, 2017.

[Liang *et al.*, 2017] Xiaodan Liang, Lisa Lee, and Eric P. Xing. Deep variation-structured reinforcement learning for visual relationship and attribute detection. In *Proceedings of CVPR*, pages 4408–4417, 2017.

[Liao *et al.*, 2017] Wentong Liao, Shuai Lin, Bodo Rosenhahn, and Michael Ying Yang. Natural language guided visual relationship detection. *CoRR*, abs/1711.06032, 2017.

[Lin *et al.*, 2014] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. In *Processings of ICLR*, 2014.

[Lin *et al.*, 2015] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of AAAI*, pages 2181–2187, 2015.

[Lu *et al.*, 2016] Cewu Lu, Ranjay Krishna, Michael S. Bernstein, and Fei-Fei Li. Visual relationship detection with language priors. In *Proceedings of ECCV*, pages 852–869, 2016.

[Maaten and Hinton, 2008] Laurens Van Der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2605):2579–2605, 2008.

[Marino *et al.*, 2017] Kenneth Marino, Ruslan Salakhutdinov, and Abhinav Gupta. The more you know: Using knowledge graphs for image classification. In *Proceedings of CVPR*, pages 20–28, 2017.

[Socher *et al.*, 2013] Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of NIPS*, pages 926–934, 2013.

[Wang *et al.*, 2014] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of AAAI*, pages 1112–1119, 2014.

[Xiao *et al.*, 2016] Han Xiao, Minlie Huang, and Xiaoyan Zhu. TransG: A generative model for knowledge graph embedding. In *Proceedings of ACL*, pages 2316–2325, 2016.

[Xu *et al.*, 2017] Danfei Xu, Yuke Zhu, Christopher Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of CVPR*, pages 5410–5419, 2017.

[Yu *et al.*, 2017] Ruichi Yu, Ang Li, Vlad I. Morariu, and Larry S. Davis. Visual relationship detection with internal and external linguistic knowledge distillation. In *Proceedings of ICCV*, pages 1068–1076, 2017.

[Zhang *et al.*, 2017a] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual translation embedding network for visual relation detection. In *Proceedings of CVPR*, pages 3107–3115, 2017.

[Zhang *et al.*, 2017b] Hanwang Zhang, Zawlin Kyaw, Jinyang Yu, and Shih-Fu Chang. PPR-FCN: weakly supervised visual relation detection via parallel pairwise R-FCN. In *Proceedings of ICCV*, pages 4243–4251, 2017.

[Zhu *et al.*, 2017] Yuke Zhu, Joseph J Lim, and Li Fei-Fei. Knowledge acquisition for visual question answering via iterative querying. In *Proceedings of CVPR*, pages 1154–1163, 2017.