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 Yet-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 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
the entities completely in an image is a time-consuming and difficult job, which makes a scene graph usually suffer from incompleteness. 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), abridged as \((h, r, t)\). 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.

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 \(\mathcal{E}\) (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 \(\mathcal{E}_i \subseteq \mathcal{E}\). Each entity \(e_{i,k} = (e_{i,k}, A_{i,k}, b_{i,k}) \in \mathcal{E}_i\) has an entity type \(e_{i,k} \in \mathcal{E}_i\), where \(k \in \{1, \ldots, |\mathcal{E}_i|\}\). Entity \(e_{i,k}\) is packed with attributes \(A_{i,k} \subseteq \mathcal{A}\) 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., \((\text{sandwich}, \{\}, b_1), \text{on}, \text{(plate, \{white\}, b_2})\). For simplicity, we write as \((\text{sandwich}, \text{on, plate})\).

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 \(\mathcal{T}_i\), \(\exists e_{i,p}, e_{i,q} \in \mathcal{E}_i \land \exists r_{1} \in \mathcal{R}_i \land \exists r_{2} \in \mathcal{R}_i\) s.t. \((e_{i,p}, r_{1}, e_{i,q}) \in \mathcal{T}_i\), where \(p \neq q\). Scene graph completion is to find new visual triples \((e_{i,p}, r_{1}, e_{i,q}) \in \mathcal{T}_i\) and then extend \(\mathcal{T}_i\) to be \(\mathcal{T}_i^g = \mathcal{T}_i \cup \{(e_{i,p}, r_{1}, e_{i,q})|e_{i,p}, e_{i,q} \in \mathcal{E}_i, r_{1} \in \mathcal{R}_i, p \neq q\}\) s.t. \(|\mathcal{T}_i^g| > |\mathcal{T}_i|\).

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

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

\[
E(h_t, r, t) = ||h_t + r - t||_2 / l_{1,2}.
\]

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

TransD. For each triple \((h_t, r, t)\), TransD assigns \(h_t, t\) and \(r\) with extra projection vectors \(h_{tp}, t_{tp}\), and \(r_{tp}\). They set two mapping matrices \(M_{h_{tp}}, M_{t_{tp}} \in \mathbb{R}^{m \times n}\) to project entities from entity space to relation space as follows:

\[
M_{h_{tp}} = r_p h_{tp}^T + \Gamma p \times n, \quad M_{t_{tp}} = r_p t_{tp}^T + \Gamma p \times n.
\]

\(M_{h_{tp}}\) and \(M_{t_{tp}}\) are determined by both entities and relations, which makes the two projection vectors interact sufficiently.
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 $\mathcal{T}$ as follow:

$$E_I(h, r, t) = \|h + r - t\|_{L_1/L_2}.$$  

(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, instantiatiated, 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 $U$ of shape $H' \times W' \times C$, where $H' = \left\lceil \frac{H}{32} \right\rceil$, $W' = \left\lceil \frac{W}{32} \right\rceil$, 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 $U$. 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 $U$ and an projected entity bounding box on $U$, we can get a local feature map of entity $V$ in convolutional style as follow:

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

(4)

where $G$ is a sampling grid of shape $Y \times X \times 2$ marking real-valued coordinates of $V$ on $U$ 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 $V_h$ and $V_t$, 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 $1 \times 1$ 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, $V_h$ (resp., $V_t$) 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 $1 \times 1$ convolutional layers $conv-R1$ and $conv-R2$ (Line 3 in Figure 2). But for layer $conv-R1$, we concatenate $V_h$ and $V_t$ 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:

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

(5)

where $x, y$ are the left-top coordinates of the bounding box and $w, h$ are its width and height (likewise for $x', y', w', h'$). 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 $v_{pr}$ 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 $M_h^v$, $M_r^v$, $M_t^v \in \mathbb{R}^{n \times n}$ ($c$ 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:

$$h_{\perp} = M_h^v M_h^t M_h^v h, \quad t_{\perp} = M_t^v M_t^r M_t^v t.$$  

(6)

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

$$r_{\perp} = M_r^v r.$$  

(7)

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

\[ Figure 2: Architecture of RLSV model. \]
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 $e_p \in \mathbb{R}^n$ (resp., $r_p$ and $a_p$), 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 $A_e$, we set the number of attributes $N_a = |A_e|$. For each $a_i \in A_e$, we construct the attribute projection matrix as follow:

$$M_p^a = a_i e_p^T + \mathbf{I}^{n \times n},$$  \hspace{1cm} (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:

$$M_p^a = \sum_{i=1}^{N_a} \beta_i M_p^a_i = \left( \sum_{i=1}^{N_a} \beta_i a_i \right) e_p^T + \mathbf{I}^{n \times n},$$  \hspace{1cm} (9)

where $\beta_i$ 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 $v_p$, and the attribute projection vector $a_i$ as follows:

$$\epsilon_i = w_h^T \text{tanh}(W_v v_p + W_a a_i),$$  \hspace{1cm} (10)

$$\beta_i = \text{softmax}(\epsilon_i) = \frac{\exp(\epsilon_i)}{\sum_{j=1}^{N_a} \exp(\epsilon_j)},$$  \hspace{1cm} (11)

where $w_h \in \mathbb{R}^n$ and $W_v, W_a \in \mathbb{R}^{n \times n}$ are learnable weights. Particularly, for an entity without any attribute, i.e., $N_a = 0$, we set $M_p^a = \mathbf{I}^{n \times n}$, which means there is no need to project onto attribute space. From Equation 8-11, we get the projection matrix $M_p^h$ (resp., $M_p^r$) 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 semantic 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 $v_p$ (resp., $v_t$) and the relation projection vector $r_p$ as follows:

$$M_p^r = r_p v_p^T + \mathbf{I}^{n \times n},$$  \hspace{1cm} (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:

$$M_p^h = v_p h_p^T + \mathbf{I}^{n \times n},$$

$$M_p^r = v_p r_p^T + \mathbf{I}^{n \times n},$$

$$M_p^t = v_p t_p^T + \mathbf{I}^{n \times n}.$$  \hspace{1cm} (13)

3.4 Objective

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

$$L = \sum_{I \in \mathcal{L}} \sum_{(h, r, t) \in \mathcal{T}_I} \sum_{(h', r', t') \in \overline{\mathcal{T}_I}} \left[ E_f(h, r, t) + \gamma - E_f(h', r', t') \right]^+, $$  \hspace{1cm} (14)

where $[x]^+ = \max(0, x)$ and $\gamma$ is a margin hyperparameter. $\overline{\mathcal{T}_I}$ stands for the negative sampled visual triple set generated from positive visual triple set $\mathcal{T}_I$. We define $\mathcal{T}_I^T$ as follow:

$$\mathcal{T}_I^T = \{(h', r, t') \in \mathcal{E} \} \cup \{(h, r, t)|t' \in \mathcal{R}, \ (h, r, t) \in \mathcal{T}_I \} \cup \{(h, r, t)|t' \in \mathcal{R}, \ (h, r, t) \in \mathcal{T}_I^T \}. $$  \hspace{1cm} (15)

The entities and relation in a visual triple can be randomly replaced by another one, but we assure $\mathcal{T}_I \cap \mathcal{T}_I^T = \emptyset$. We constrain the norms of both original and projected embeddings of a visual triple as $\|h\|_2 \leq 1$, $\|t\|_2 \leq 1$, $\|r\|_2 \leq 1$, $\|h_+\|_2 \leq 1$, $\|t_+\|_2 \leq 1$, and $\|r_+\|_2 \leq 1$.

3.5 Optimization and Implementation Details

The learnable parameter set of RLSV model can be formalized as $\theta = (E, R, E_p, R_p, A_p, W)$, where $E$ (resp., $R$) represents the entity (resp., relation) embeddings; $E_p, R_p$, and $A_p$ represent the projection vector sets of entity, relation, and attribute; and $W$ 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]. $E, R, E_p, R_p,$ and $A_p$ 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 epochs for the model to converge.

4 Experiments

We evaluate our model with two typical scene graph completion tasks: 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.

2All 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: Qualitative examples of relation prediction. On the left side, we show an image with a testing visual triple from VRD and VG. On 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.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Train</th>
<th>#Valid</th>
<th>#Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>VRD</td>
<td>26,057</td>
<td>5,641</td>
<td>5,641</td>
</tr>
<tr>
<td>VG</td>
<td>657,404</td>
<td>141,627</td>
<td>141,627</td>
</tr>
</tbody>
</table>

Table 1: Statistics of datasets

<table>
<thead>
<tr>
<th>Element</th>
<th>Relation</th>
<th>Head</th>
<th>Tail</th>
</tr>
</thead>
<tbody>
<tr>
<td>VRD</td>
<td>70.00</td>
<td>8.56</td>
<td>8.56</td>
</tr>
<tr>
<td></td>
<td>69.77</td>
<td>8.27</td>
<td>8.11</td>
</tr>
<tr>
<td>VG</td>
<td>100.00</td>
<td>16.81</td>
<td>16.81</td>
</tr>
<tr>
<td></td>
<td>99.90</td>
<td>15.97</td>
<td>16.07</td>
</tr>
</tbody>
</table>

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 \( \text{conv-EI} \) and \( \text{conv-R1} \). The number 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 \( 1 \times 10^{-3} \) and descends through iterations until \( 6.25 \times 10^{-5} \).

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 \( \text{conv-R2} \) layer with \( \text{softmax} \) 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 \( \beta_i = \frac{1}{n} \) and replace \( v_p^h, v_p^t, h_p, t_p \) in Equation 12 respectively.
- Generating scene graph only from images: VTransE and PPR-FCN, which are modified for our settings.
- 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...
3 and we ranked all scores of replaced visual triples in ascending order. Specially, for RLSV-V, VTransE and PPR-FCN, we used the input of the classifier as score. It is the “raw” setting order. Specially, for RLSV-V, VTransE and PPR-FCN, we ranked all scores of replaced visual triples in ascending order. We assured that the number of positive and negative visual triples is equal.

### 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-V-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 “fit” 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 \(\sigma_r\), then it is positive, and vice versa. The threshold \(\sigma_r\) 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 results of 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 results of VTransE and PPR-FCN.
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 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

This paper was supported by the National Natural Science Foundation of China (No. 61573386), Guangdong Province Natural Science Foundation (No. 2016A030313292), Guangdong Province Science and Technology Plan projects (No. 2016B030305007 and No. 2017B010110011), Guangzhou Science and Technology Project (No. 201804010435), and Sun Yat-sen University Young Teachers Cultivation Project (No. 16lgpy40) and GF Cultivation Project.
References


