Scene Graph Refinement with Commonsense Knowledge

 # 228

Abstract

This paper focuses on scene graph refinement which aims at refining the scene graph generated by existing scene graph generation models. Most of the scene graph generation works are only data-driven fashion and take Recall@50 and Recall@100 as evaluation metrics, which leads to low-quality predictions with much noise. To deal with the problem, we propose a method that introduces commonsense knowledge as post-processing to refine the scene graph. We formulate scene graph refinement task as an integer linear programming (ILP) problem with the objective function generated from existing models and the constraints translated from commonsense knowledge, which can result in a number of preferred visual triples. Furthermore, we utilize the bounding boxes of location information in each visual triple to optimize our approach. Experimental results on two datasets, VRD and VG, demonstrate that our approach outperforms baseline methods in both tasks, which proves the significance of introducing commonsense knowledge to scene graph refinement.

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 $(subject, predicate, object)$ in which a $subject/object$ is grounded with a bounding box in its corresponding images, and a $predicate$ is an edge from the $subject$ to the $object$ (see example in Figure 1). Scene graph is extremely useful for many Artificial Intelligence related applications, such as image caption [Anderson *et al.*, 2016; Liu *et al.*, 2018; Zhao *et al.*, 2018] and visual question-answering [Dong *et al.*, 2015; Lin *et al.*, 2018; Song *et al.*, 2018]. Hence, scene graph has attracted increasing attention.

Scene graph generation has become an increasingly important task. [Krishna *et al.*, 2017] collected scene graphs by crowdsourced dense image annotations. There are also many automated models that depend on object detection. Most of the previous works are only data-driven fashion, such as LP [Lu *et al.*, 2016], VTransE [Zhang *et al.*, 2017a], PPR-FCN [Zhang *et al.*, 2017a], DR-Net [Dai *et al.*, 2017], LK [Yu *et al.*, 2017], VRL [Liang *et al.*, 2017], DSL [Zhu and Jiang, 2018], Motif [Zellers *et al.*, 2018], DSR [Liang *et al.*, 2018], and MP [Xu *et al.*, 2017], *et al.*

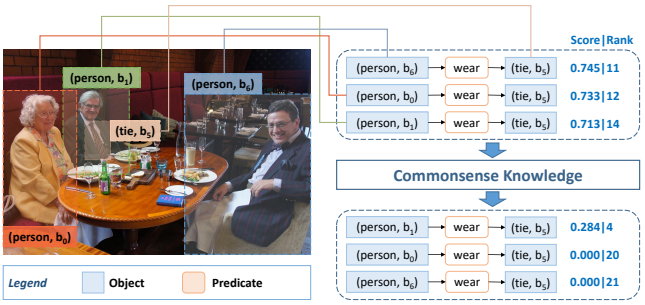

Figure 1: An example image from VRD. DSR model result shows on the upper blue dashed box. The result refined by commonsense knowledge shows on the below blue dashed box.

Most of the recent scene graph generation tasks take Recall@50 and Recall@100 as the evaluation metric. However, these evaluation metrics cannot demonstrate the experimental results. Though there are many false visual triples in the generated scene graph, the score of Recall@50 or Recall@100 are still high. For example, DSR [Liang *et al.*, 2018] predicate classification Recall@50 in VRD is 86.01% while Recall@5 is only 13.74%. So it is clear that the predictions are low-quality and contain many false visual triples. As an example in Figure 1, the scene graph generation model predicts (tie, b_5) has relation $wear$ with $(person, b_0)$, $(person, b_1)$, $(person, b_6)$ and their scores and rank are similar. Actually, only $((person, b_1), wear, (tie, b_5))$ is the correct visual triple. Besides, most of the images in VRD and VG datasets have been labeled with only twenty to thirty visual triples in ground truth. Hence, we should use Recall@5 or Recall@10 to evaluate the predictions.

To deal with the above problem, we propose a method that introduces commonsense knowledge as post-processing to refine the scene graph. We introduce commonsense knowledge and the bounding box location information into scene graph refinement. Rule is one of the most important expression method of commonsense knowledge. Commonsense knowledge which is expressed by rules have been demonstrated to play a pivotal role in inference [Jiang *et al.*, 2012; Pujara *et al.*, 2013], as well as in knowledge base completion

[Wang *et al.*, 2015]. Commonsense knowledge will give the prediction some constraints, which can refine the prediction.

As we know, $wear$ is *one-to-many* type relation. For example, there is a tie and several $person$ in the image in Figure 1. Obviously, one tie can only related with one $person$, such as $((person, b_1), wear, (tie, b_5))$. Other relationships should be filtered. And then we obtain more reasonable relationship scores and ranks, which improve the relationship $((person, b_1), wear, (tie, b_5))$ score and reduce others. The use of commonsense knowledge greatly reduces the score of false relationships and significantly enhances inference accuracy. Therefore, commonsense knowledge is of critical significance to scene graph generation.

Furthermore, the utilization of commonsense knowledge can not enhance the score of truth relationships in some cases. As we know, object location information is conducive to judge the relationships between two objects. For instance, $(person, b_1)$ is nearer with (tie, b_5) than $(person, b_0)$ and $(person, b_6)$ in Figure 1. It is obvious that (tie, b_5) has relation with $(person, b_1)$. So the object location information should be considered.

There are two challenges to apply commonsense knowledge to the scene graph refinement: (1) how to extract commonsense knowledge from various relationships; (2) how to use transferred rules to filter wrong relationships and refine predictions.

To solve the problems above, we propose a novel scene graph refinement approach that refines scene graph using commonsense knowledge. In order to transfer relationships to commonsense knowledge, we present a new approach that formulates scene graph generation as an integer linear programming (ILP) problem. The object function is the aggregated plausibility of all candidate facts, predicted by a specific scene graph generation model; and the constraints are translated from commonsense knowledge. Solving the ILP problem results in a number of relationships which are the most preferred by the existing models or comply with all the rules. Furthermore, we propose a simple approach to reduce the wrong relationships scores by utilizing the bounding box location information of objects.

Specifically, our contributions are as follows:

- To the best of our knowledge, it is the first attempt to introduce commonsense knowledge rules to scene graph refinement .
- We formulate scene graph refinement as an ILP problem, with the objective function generated from existing models and the constraints translated from rules.
- Our method introduces commonsense knowledge and the bounding box location information to refine scene graph generated by scene graph generation models.

2 Preliminary

Scene graph. For *scene graph generation* task, the goal is to generate a visually-grounded scene graph describing objects and their relationships in the image. Following [Wan *et al.*, 2018], we assume that all images are in a finite set $\mathcal{I}$. All object classes in $\mathcal{I}$ are in a finite set $\mathcal{C}$. All predicates in $\mathcal{I}$ are in a finite set $\mathcal{P}$. The scene graph $\mathcal{G}$ of each image consists of:

- a set $B = \{b_1, ..., b_n\}$ of bounding boxes, $b_i \in \mathcal{R}^4$,
- a corresponding set $O = \{o_1, ..., o_n\}$ of objects, assigning a class label $o_i \in \mathcal{C}$ to each b_i, and
- a set $R = \{r_1, ..., r_m\}$ of visual relationships. Each relationship r_i is a triple in the form of $((o_i, b_i), p, (o_j, b_j))$, where $p \in \mathcal{P}$.

Scene graph generation. Scene graph generation is to automatically generate a scene graph to represent an image. [Lu *et al.*, 2016] proposed a typical model with language prior. In their method, pairs of detected objects are fed to a classifier, which combines appearance features and a language prior for relationship recognition. [Zhang *et al.*, 2017a] applied a translation embedding model to place objects in a low-dimensional relation space so that a relation can be modeled as a simple vector translation. [Zhang *et al.*, 2017b] used a parallel FCN architecture and a position-role-sensitive score map to tackle the task. [Dai *et al.*, 2017] exploited the statistical dependencies between objects and their relationships. [Zhu and Jiang, 2018] introduced deep structured learning for visual relationship detection. [Liang *et al.*, 2018] proposed a deep neural network framework with the structural ranking loss. Although these methods have achieved some effects, such strategies would be met with a fundamental difficulty that the image information may cause some confusions and thus the predicates cannot be predicted reliably. [Yu *et al.*, 2017] proposed a teacher-student knowledge distillation framework making use of internal and external knowledge. [Liang *et al.*, 2017] constructed a directed semantic action graph and used deep variation-structured reinforcement to predict visual relationships. While these models got some improvements by combining external and internal knowledge, they brought a lot of noise in the meantime. [Xu *et al.*, 2017] proposed an end-to-end model that solves the scene graph inference problem by RNNs and learns to iteratively improves its predictions by message passing. MotifNet [Zellers *et al.*, 2018] presents new quantitative an analysis of VG dataset showing that motifs are prevalent. They introduce Stacked Motif Networks, which is a new architecture designed to capture higher-order motifs in the scene graph. The main drawback of MotifNet is that the network is so complex. The methods mentioned above only exploit information from the image, and do not make use of commonsense knowledge rules.

Commonsense knowledge. Rule is one of the most important expression of commonsense knowledge. Rules, particularly logical rules, have been studied extensively in MRF-based (Markov Random Field) knowledge base completion model, represented in first-order logic [Richardson and Domingos, 2006] and probabilistic soft logic [Bröcheler *et al.*, 2010]. Moreover, [Wang *et al.*, 2015] introduce physical and logical rules to impose restraints on knowledge base completion. In this paper, we apply the first one as commonsense knowledge rules to scene graph refinement task.

Integer linear programming. Integer linear programming (ILP) refers to constrained optimization where both the objective and constraints are linear equations with integer vari-

ables. It has been widely used in many different fields, such as knowledge base completion [Wang *et al.*, 2015]. This paper employs ILP to integrate scene graph generation models and rules in a framework for scene graph generation.

3 Our Approach

In this section, we give some definitions and introduce our approach, which aims at refining relationships predicted by scene graph generation model.

3.1 Task Definition and Overview

For *scene graph generation* task, the goal is to generate a visually-grounded scene graph describing objects and their relationships in the image. We assume that all images are in a finite set $\mathcal{I}$. All object classes in $\mathcal{I}$ are in a finite set $\mathcal{C}$. All predicates in $\mathcal{I}$ are in a finite set $\mathcal{P}$. The scene graph $\mathcal{G}$ of each image consists of:

- a set $B = \{b_1, ..., b_n\}$ of bounding boxes, $b_i \in \mathcal{R}^4$,
- a corresponding set $O = \{o_1, ..., o_n\}$ of objects, assigning a class label $o_i \in \mathcal{C}$ to each b_i, and
- a set $R = \{r_1, ..., r_m\}$ of visual relationships. Each relationship r_i is a triple in the form of $((o_i, b_i), p, (o_j, b_j))$, where $p \in \mathcal{P}$.

Our problem is formulated as follows: Given an image $I \in \mathcal{I}$ and its scene graph $\mathcal{G}_I = \{B_I, O_I, R_I\}$ generated by scene graph generation model, our goal is to refine R_I using commonsense knowledge.

The overview of our approach is shown in Figure 2. Detailed introductions to different components will be given in the following sections. The entire process can be summarized as the following steps: (1) employ scene graph generation model to capture the objects in the image and their pairwise relationships; (2) introduce rules to impose constraints on those relationships; (3) integrate the first two components by ILP, with the objective function generated from the scene graph generation model and the constraints translated from the rules. In this way, relationships will have the highest accuracy predicted by scene graph generation model, and at the same time comply with all the rules.

3.2 Scene Graph Generation Model

We employ two scene graph generation models: Deep Structural Ranking (DSR) model [Liang *et al.*, 2018] and Message Passing (MP) model [Xu *et al.*, 2017].

DSR. Given a relationship $r_k = ((o_i, b_i), p, (o_j, b_j))$ in an image I, where o_i, b_i, o_j, b_j are captured by the object detector (*e.g.* Faster R-CNN) [Ren *et al.*, 2015] or from ground truth. DSR uses the following score function to indicate the accuracy of the relationship:

$$\Phi(I, r_k) = w_p^T f(I, (o_i, b_i), (o_j, b_j)) \quad (1)$$

where w_p denotes the parameters to be learned for predicate p, and $f(I, (o_i, b_i), (o_j, b_j))$ denotes the fuse function of visual appearance, spatial location and sematic embedding features detailed in [Zhu and Jiang, 2018] for relationship instance r_k. The parameters w_p is learned by minimize the structural ranking loss:

$$L = \sum_{r \in R} \sum_{r' \in R'} [\triangle(r, r') + \Phi(I, r') - \Phi(I, r)]_+ \quad (2)$$

where R' is the set of relationship instance that are not in annotations, $[\cdot]_+ = max(0, \cdot)$ and $\triangle(\cdot, \cdot)$ is a margin that distinct the different relationship instances.

MP. They regard objects and predicates as objects nodes v^O and predicates nodes v^R respectively in a virtual graph $G = (V = V^O \cup V^R, E)$, where $v^O \in V^O$ denotes object, $v^R \in V^R$ denotes predicates, and edge $e = (v_i^O, v_j^O) \cup (v_i^O, v_{ij}^R) \cup (v_j^O, v_{ij}^R) \in E$ means that if object i and j are related, there are edges between v_i^O and v_j^O, v_i^O and v_{ij}^R, as well as v_j^O and v_{ij}^R. Each node has its own feature and broadcasts its message to neighbors to instruct them obtain better features. Supposing f_i^O and f_j^O are features of two object candidates associated with v_i^O and v_j^O, and f_{ij}^R represents the relationships feature associated with v_{ij}^R, the message passing procedure can be written as:

$$m_i^O = G^O\left(\sum_{j \in N_i^O} M^{O \to O}(f_j^O), \sum_{j \in N_i^O} M^{R \to O}(f_{ij}^R)\right), \quad (3)$$

$$m_{ij}^R = G^R\Big(M^{O \to R}(f_i^O), M^{O \to R}(f_j^O)\Big), \quad (4)$$

$$f_i^O \leftarrow U^O(f_i^O, m_i^O), \quad (5)$$

$$f_{ij}^R \leftarrow U^R(f_{ij}^R, m_{ij}^R), \quad (6)$$

where m_i and m_{ij}^R denote message received by node v_i^O and v_{ij}^R respectively. N_i^O stands for neighbors of v_i^O. $M^{O \to O}$, and $M^{O \to R}$ are message processing functions. G^O and G_R represent gathering functions which integrate message from sources. U^O and U^R are update functions for object and predicate. After message passing process, the refined features could be used to make predictions.

After we obtain the score of each relationship, we find that some of the wrong relationships scored higher than correct relationships. Therefore, we further propose a simple approach to reduce the score of wrong relationships by utilizing the bounding boxes location information. First, we use the following function to map the score to a continuous truth value which lies in the range of (0,1):

$$\sigma(r_k) = 1/(1 + \exp(-S_{raw}(r_k))), \quad (7)$$

where $S_{raw}(r_k)$ is the corresponding score of r_k calculated by the score function of scene graph generation model. Then, we use the following formula to calculate a distance score:

$$S_{dis}(r_k) = \phi(1/dis(r_k)), \quad (8)$$

where $\phi(x) = (x - min)/(max - min)$ denotes the min-max normalization function, and $dis(r_k)$ represents the distance between the center points of b_i and b_j. The total score is:

$$S_{total}(r_k) = (1 - \alpha)S_{raw}(r_k) + \alpha S_{dis}(r_k), \quad (9)$$

where α is the weighting factor for distance score.

3.3 Commonsense Knowledge

Here, we introduce two rules to represent commonsense knowledge, imposing constraints on relationships.

Rule 1 (predicate expectation). The $predicate$ connecting an object class pair should be of certain types. For instance,

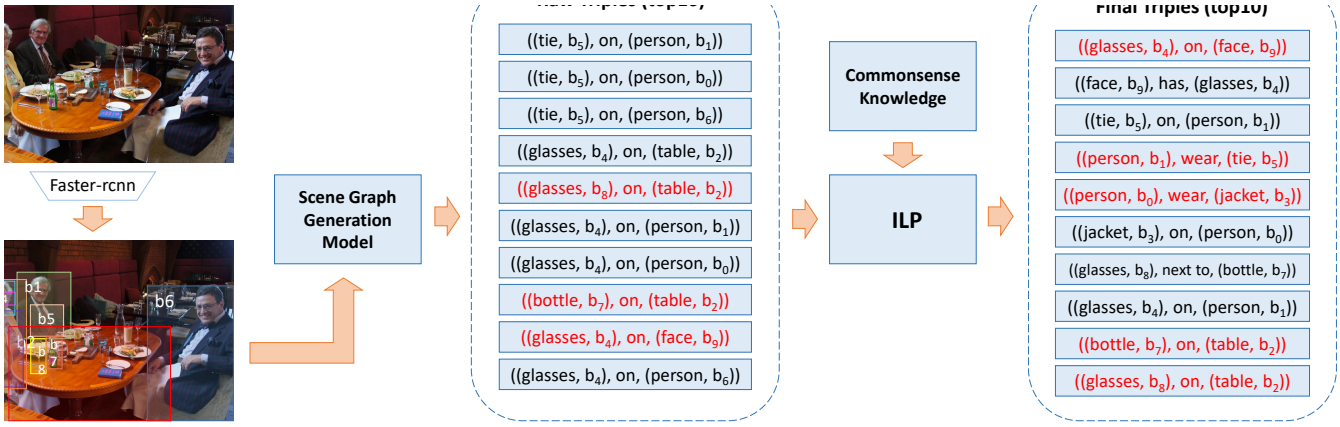

Figure 2: The overview of our approach. Given an image, it first employs Faster R-CNN to locate individual objects. Then employing the existing scene graph generation model to capture the relationships between objects in the image and obtain raw triples. After that, we translate rules into constraints and integrate it with raw triples by ILP. Finally, we obtain the final triples. Relationships that exist in ground truth are marked red. (Here we employ DSR as the scene graph generation model.)

the $predicate$ between bus and $tree$ can not be eat. As a matter of fact, only a few kinds of predicates can connect an object class pair despite that a dataset usually has dozens of predicates. For example, only six kinds of predicates between bus and $tree$ appeared in the training set of VRD dataset, which are $in\ the\ front\ of$, $behind$, $on\ the\ left\ of$, $near$, $under$, and $next\ to$. We can see that all the predicates are geometric predicates, and there is no semantic predicates such as eat or $hold$.

Rule 2 (at-most-one constraint). Following the same classification method[1] in [Bordes *et al.*, 2013] , we break down predicates into 4 categories: one-to-one, one-to-many, many-to-one, many-to-many. However, some predicates in the VG dataset will be assigned to the wrong category, such as $wear$, which is a one-to-many predicate but will be assigned many-to-many based on this method. Therefore, we made some artificial corrections. The specifications are shown in Table 1. For one-to-one predicates, the connected object pair can take at most one subject and object; for one-to-many predicates, the object can connect at most one subject; for many-to-one predicates, the subject can connect at most one object. As an example shown in Figure 1, $wear$ is a one-to-many predicate. Given an object such as (tie, b_5), there exists at most one subject for which the relationship is true.

These rules contain rich prior knowledge, and can greatly improve the performance of scene graph refinement.

3.4 Integrating by Integer Linear Programming

We aggregate the above two components and formulate our task as an ILP problem. For each relationship $r = ((o_i, b_i), p_k, (o_j, b_j))$[2], we use $w_{ij}^k = f(r)$ to represent the accuracy predicted by a scene graph generation model, and introduce a Boolean decision variable x_{ij}^k to indicate whether the relationship is true or false. Our aim is then to find the best assignment to the decision variables, maximizing the overall accuracy and complying with all the constraints. The ILP problem is given in the following formula:

$$\max_{\{x_{ij}^k\}} \sum_k \sum_i \sum_j w_{ij}^k x_{ij}^k, \tag{10}$$

where x_{ij}^k should satisfy all the constraints. According to the Rule1, we have the following constraint:

$$x_{ij}^k = 0, \quad \forall k \notin \mathcal{D}_{ij}, \tag{11}$$

where $\mathcal{D}_{ij}$ including all kinds of predicates that connect a certain object class pair (o_i, o_j). And we can translate Rule2 to other constraints:

$$\sum_i x_{ij}^k \leq 1, \quad \forall k \in \mathcal{P}_{o-m}, \forall j, \tag{12}$$

$$\sum_j x_{ij}^k \leq 1, \quad \forall k \in \mathcal{P}_{m-o}, \forall i, \tag{13}$$

$$\sum_i x_{ij}^k \leq 1, \sum_j x_{ij}^k \leq 1, \quad \forall k \in \mathcal{P}_{o-o}, \forall i, \forall j, \tag{14}$$

where $\mathcal{P}_{o-m}/\mathcal{P}_{m-o}/\mathcal{P}_{o-o}$ refers to the set of one-to-many/many-to-one/one-to-one predicates.

This ILP problem is decomposable for every k and can be simplified as follows.

In case $k \in \mathcal{P}_{o-m}$, x_{ij}^k can be directly computed by

$$x_{ij}^k = \begin{cases} 1 & \text{if } k \in \mathcal{D}_{ij} \text{ and } j = \arg\max_j w_{ij}^k \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

[1]For each predicate, we compute the averaged number of *subject types* o_s or *object types* o_o appearing in the training set, given a pair *(p, o_o)* or a pair *(o_s, p)*. If the averaged number is smaller than 1.5, we labeled the argument as "one" and "many" otherwise. For example, a predicate has an average of 1.4 *subject types* per *object types* and of 3.5 *object types* per *subject types* is classified as one-to-many.

[2]For simplicity, we write (o_i, b_i), (o_j, b_j) and p_k as i, j, k respectively.

Table 1: Relation types of VRD, VG, and VG*

relation type	VRD		VG		VG*	
	num	example	num	example	num	example
one-to-one	11	hit, face, kick	22	attach to, contain, face	6	attach to, for, of
one-to-many	26	wear, use, at	31	hold, have, touch	13	holding, has, playing
many-to-one	9	cover, eat, against	14	pull, from, catch	14	across, against, belonging to
many-to-many	24	on, in, above	33	on, near, over	17	above, along, and

Table 2: Experiment-VRD

Model	Predicate Cls.			
	R@5	R@10	R@15	R@20
DSR	13.74	23.43	30.50	35.97
DSR+ours	**16.06**	**28.24**	**34.98**	**42.27**

Table 3: Experiment-VG

Model	Predicate Cls.			
	R@5	R@10	R@15	R@20
DSR	6.05	10.38	13.92	17.04
DSR+ours	**8.27**	**13.91**	**18.34**	**21.82**

In case $k \in \mathcal{P}_{m-o}$, x_{ij}^k can be directly computed by

$$x_{ij}^k = \begin{cases} 1 & \text{if } k \in \mathcal{D}_{ij} \text{ and } i = \arg\max_i w_{ij}^k \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

In case $k \in \mathcal{P}_{o-o} \cap \mathcal{D}_{ij}$, x_{ij}^k can be computed by another ILP problem that are much smaller than the original one, i.e., by maximizing

$$\sum_i \sum_j w_{ij}^k x_{ij}^k \tag{17}$$

under the following constraints.

$$\sum_i x_{ij}^k \leq 1 \qquad \forall j \tag{18}$$

$$\sum_j x_{ij}^k \leq 1 \qquad \forall i \tag{19}$$

In other cases, x_{ij}^k can be directly computed by

$$x_{ij}^k = \begin{cases} 1 & \text{if } k \in \mathcal{D}_{ij} \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

4 Experiments

In the following subsections, we firstly introduce experimental settings including datasets, evaluation metrics, and implementation details. Then, we show the experiment results.

4.1 Datasets

We evaluate our approach on Visual Relationship Datasets (VRD) [Lu *et al.*, 2016] and Visual Genome(VG) [Krishna *et al.*, 2017]. The specifications of these datasets are shown in Table 4.

VRD contains 5,000 images with 100 object classes and 70 predicates. VRD contains 37,993 relation annotations with 6,672 type triples in total. Following the same train/test split as in [Lu *et al.*, 2016], we split images into two sets, 4,000 images for training and 1,000 for testing.

Table 4: Statistics of datasets

Dataset	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{C}\rvert$	$\lvert\mathcal{P}\rvert$	$\lvert\mathcal{I}_{train}\rvert$	$\lvert\mathcal{I}_{test}\rvert$
VRD	5,000	100	70	4,000	1,000
VG	99,658	200	100	73,801	25,857
VG*	108,077	150	50	76,201	31,876

VG contains 99,658 images with 200 object classes and 100 predicates. Totally, VG contains 1,174,692 relation annotations with 19,237 type triples. Following the experiments in [Zhang *et al.*, 2017a], we split the data into 73,801 for training and 25,857 for testing.

VG* dataset is from [Xu *et al.*, 2017]. The dataset contains an average of 25 distinct objects and 22 relationships per image. In this experiment, we use the most frequent 150 object categories and 50 predicates for evaluation.

4.2 Evaluation Metrics

As mentioned before, scene graph refinement aims at refining the scene graph generated by scene graph generation models. To evaluate our approach, we prefer to analyze all the models in predicate classification which given the boxes and labels of objects in an image. We further analyze our approach in scene graph classification without given labels. We take the original model without rules as baseline in our experiments.

- **Predicate Classification** (Predicate Cls.) task is to predict the predicate of all pairwise relationships of a set of ground truth boxes and labels.
- **Scene Graph Classification** (SG Cls.) task is to predict the predicate as well as the object labels of the subject and the object in every pairwise relationship given a set of ground truth boxes.

Because of the incompleteness of annotated scene graphs, mean average precision (mAP) would falsely penalize positive predictions on unlabeled relationships. Therefore we use Recall@K (R@K) metrics for comparison.

4.3 Implementation Details

We use the code released by [Liang *et al.*, 2018] and [Xu *et al.*, 2017] for DSR and MP respectively. Both DSR and MP adopt pretrained VGG-16 network to extract visual features from images. During training, the weights of all convolutional layers are fixed in VGG-16. We then incorporate rules into the two models with the same parameters setting using ILP, compared with the models without rules.

4.4 Comparative Results

Table 2, Table 3, and Table 5 show the results on the VRD, VG, and VG* datasets, respectively. We use R@5, R@10,

Table 5: Experiment-VG*

Model	Predicate Cls.				SG Cls.			
	R@5	R@10	R@15	R@20	R@5	R@10	R@15	R@20
MP	33.56	47.00	54.90	60.32	20.72	28.11	32.29	30.95
MP+ours	**36.74**	**49.28**	**57.32**	**63.58**	**21.95**	**30.73**	**36.17**	**39.27**

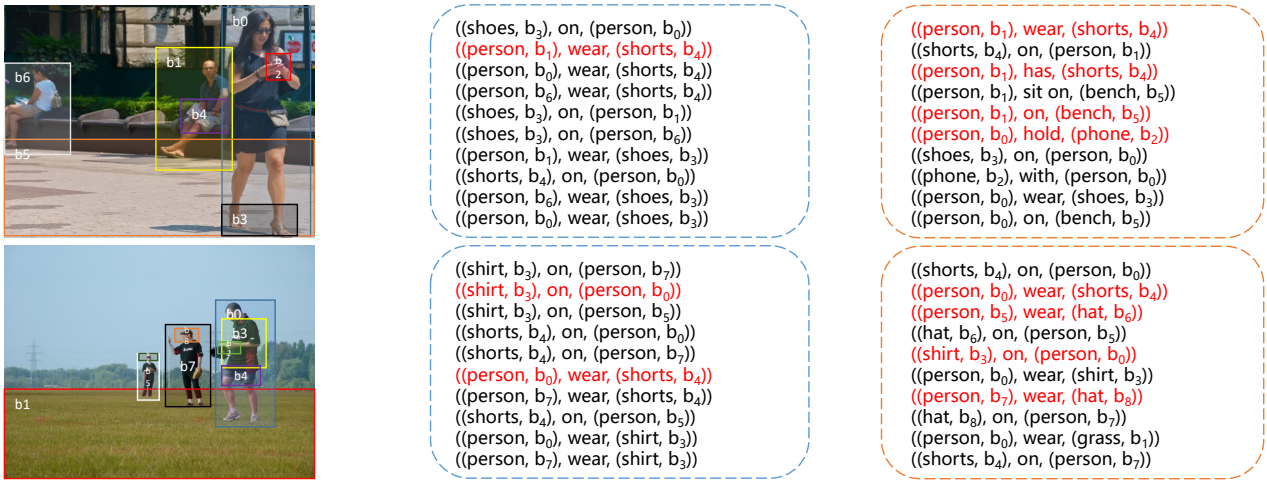

Figure 3: Qualitative examples from our approach in the predicate classification setting. The raw triples are predicted by DSR, which show on the blue dashed box. The results refined by rules show on the orange dashed box. Triples that exist in ground truth are marked red.

R@15, and R@20 as evaluation metrics. From the results, we can conclude that in predicate classification and scene graph classification tasks, models by incorporating rules improve the performance on both VRD and VG datasets. In predicate classification task, DSR incorporation of rules improves averaged about 3.5% and 3% respectively in VRD and VG. In both predicate classification and scene graph classification tasks, MP incorporation of rules also has improved. Particularly, MP incorporation of rules achieves 39.27 at R@20, which is over 26.88% relative improvement over the original MP model. It is indicated that commonsense knowledge plays an important role in scene graph refinement task.

4.5 Qualitative Results

Qualitative examples of our approach, shown in Figure 3, suggest that our approach is able to refine relationships. As you can see from these two images, before using rules, only 1 (the second image) or 2 (the first image) of the top 10 raw triples exist in the ground truth, while our method increased the number of correct triples to 4.

Besides, it is obvious that quite a few predicates of the top 10 raw triples are *on* (the top 10 raw triples of these two images contain 5 and 4 relationships with predicate *on*, respectively), because relationships with *on* are the majority in the training set. This result shows that the scene graph generation model tends to predict relationships with high-frequency predicates. By using rules, this phenomenon will be alleviated because we can improve the ranking of relationships with other predicates.

Furthermore, visual inspection of the results suggests that the method works even better than the quantitative results would imply, since many seemingly correct relationships do not exist in the ground truth. For example, in the second image, $((shorts, b_4), on, (person, b_1))$ is a correct relationship since $((person, b_1), wear, (shorts, b_4))$ exist in the ground truth, and we can also easily judge that $((person, b_0), wear, (shoes, b_3))$ is correct.

5 Conclusion

In this paper, we have proposed a novel scene graph refinement approach which introduces commonsense knowledge and bounding box location information to refine the scene graph. It formulates scene graph generation as an integer linear programming (ILP) problem, with the objective function generated from existing models and the constraints translated from rules. Besides, we have proposed a simple approach to reduce the score of false relationships using the bounding box location information of objects. Experimental results have clearly demonstrated that our approach has good potential in scene graph generation.

In the future, we plan to explore how to process many-to-many predicate and transfer it to constraints. Furthermore, we will try to introduce commonsense knowledge during the training process of scene graph refinement.

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.

[Bröcheler *et al.*, 2010] Matthias Bröcheler, Lilyana Mihalkova, and Lise Getoor. Probabilistic similarity logic. In *Proceedings of UAI*, pages 73–82, 2010.

[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.

[Dong *et al.*, 2015] Li Dong, Furu Wei, Ming Zhou, and Ke Xu. Question answering over freebase with multi-column convolutional neural networks. In *Proceedings of ACL*, pages 260–269, 2015.

[Jiang *et al.*, 2012] Shangpu Jiang, Daniel Lowd, and Dejing Dou. Learning to refine an automatically extracted knowledge base using markov logic. In *Proceedings of ICDM*, pages 912–917, 2012.

[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.

[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 crowdsourced dense image annotations. *Proceedings of IJCV*, 123(1):32–73, 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.

[Liang *et al.*, 2018] Kongming Liang, Yuhong Guo, Hong Chang, and Xilin Chen. Visual relationship detection with deep structural ranking. In *Proceedings of AAAI*, 2018.

[Lin *et al.*, 2018] Yuetan Lin, Zhangyang Pang, Donghui Wang, and Yueting Zhuang. Feature enhancement in attention for visual question answering. In *Proceedings of IJCAI*, pages 4216–4222, 2018.

[Liu *et al.*, 2018] Anan Liu, Ning Xu, Hanwang Zhang, Weizhi Nie, Yuting Su, and Yongdong Zhang. Multi-level policy and reward reinforcement learning for image captioning. In *Proceedings of IJCAI*, pages 821–827, 2018.

[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.

[Pujara *et al.*, 2013] Jay Pujara, Hui Miao, Lise Getoor, and William W. Cohen. Knowledge graph identification. In *Proceedings of ISWC*, pages 542–557, 2013.

[Ren *et al.*, 2015] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *Proceedings of NIPS*, pages 91–99, 2015.

[Richardson and Domingos, 2006] Matthew Richardson and Pedro M. Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.

[Song *et al.*, 2018] Jingkuan Song, Pengpeng Zeng, Lianli Gao, and Heng Tao Shen. From pixels to objects: Cubic visual attention for visual question answering. In *Proceedings of IJCAI*, pages 906–912, 2018.

[Wan *et al.*, 2018] Hai Wan, Yonghao Luo, Bo Peng, and Wei-Shi Zheng. Representation learning for scene graph completion via jointly structural and visual embedding. In *Proceedings of IJCAI*, pages 949–956, 2018.

[Wang *et al.*, 2015] Quan Wang, Bin Wang, and Li Guo. Knowledge base completion using embeddings and rules. In *Proceedings of IJCAI*, pages 1859–1866, 2015.

[Xu *et al.*, 2017] Danfei Xu, Yuke Zhu, Christopher B. Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of CVPR*, pages 3097–3106, 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.

[Zellers *et al.*, 2018] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of CVPR*, pages 5831–5840, 2018.

[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.

[Zhao *et al.*, 2018] Wei Zhao, Benyou Wang, Jianbo Ye, Min Yang, Zhou Zhao, Ruotian Luo, and Yu Qiao. A multi-task learning approach for image captioning. In *Proceedings of IJCAI*, pages 1205–1211, 2018.

[Zhu and Jiang, 2018] Yaohui Zhu and Shuqiang Jiang. Deep structured learning for visual relationship detection. In *Proceedings of AAAI*, 2018.