Zero-Shot Scene Graph Relation Prediction through Commonsense Knowledge Integration

Xuan Kan, Hejie Cui, and Carl Yang*

Department of Computer Science, Emory University
{xuan.kan, hejie.cui, j.carlyang}@emory.edu

Abstract. Relation prediction among entities in images is an important step in scene graph generation (SGG), which further impacts various visual understanding and reasoning tasks. Existing SGG frameworks, however, require heavy training yet are incapable of modeling unseen (*i.e.*, zero-shot) triplets. In this work, we stress that such incapability is due to the lack of commonsense reasoning, *i.e.*, the ability to associate similar entities and infer similar relations based on general understanding of the world. To fill this gap, we propose **C**omm**O**nsense-integr**A**ted s**C**ene grap**H** r**E**lation p**R**ediction (**COACHER**), a framework to integrate commonsense knowledge for SGG, especially for zero-shot relation prediction. Specifically, we develop novel graph mining pipelines to model the neighborhoods and paths around entities in an external commonsense knowledge graph, and integrate them on top of state-of-the-art SGG frameworks. Extensive quantitative evaluations and qualitative case studies on both original and manipulated datasets from Visual Genome demonstrate the effectiveness of our proposed approach. The code is available at `https://github.com/Wayfear/Coacher`.

Keywords: Scene graph generation · Relation prediction · Zero-shot · Commonsense · Knowledge graph · Reasoning · Graph mining

1 Introduction

With the unprecedented advances of computer vision, visual understanding and reasoning tasks such as Image Captioning and Visual Question Answer (VQA) have attracted increasing interest recently. Scene graph generation (SGG), which predicts all relations between detected entities from images, distills visual information in structural understanding. With clear semantics of entities and relations, scene graphs are widely used for various downstream tasks like VQA [7, 22, 36], image captioning [4, 31, 32], and image generation [9–11].

A critical and challenging step in SGG is *relation prediction*. A relation instance in scene graph is defined as a triplet ⟨*subject, relation, object*⟩. Given two detected entities, which relation exists between them is predicted based on the probability score from the learned relation prediction model. However, most of the existing scene graph generation models rely on heavy training to memorize

* Corresponding Author.

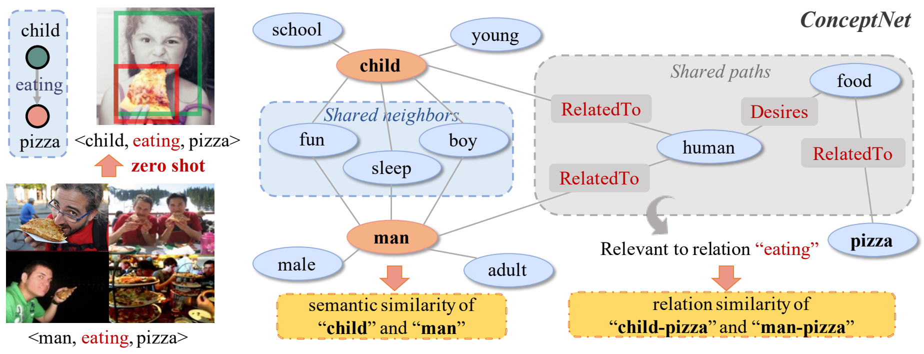


Fig. 1. Toy example of zero-shot relation prediction in scene graph generation and insights about commonsense knowledge integration.

seen triplets, which limits their utility in reality, because many real triples are never seen during training.

Challenge: Performance deterioration on zero-shot triplets. When evaluating the performance of relation prediction model in practice, there are two types of triplets, the ones seen in the training data and the ones unseen. Those unseen ones are called *zero-shot* triplets. As shown in Figure 1, triplet ⟨*man, eating, pizza*⟩ is observed in the training data. If this triplet appears again in the testing phase, then it is called a non-zero-shot triplet. In contrast, a triplet ⟨*child, eating, pizza*⟩ with a new entity-relation combination not observed in training data is called a zero-shot triplet.

Although several existing scene graph generation methods have achieved decent performance on the whole testing data, little analysis has been done on the performance on zero-shot triplets. Unfortunately, based on our preliminary analysis (Section 2.1), the performance of existing methods degrades remarkably when solely tested on zero-shot triplets, while many of such triplets are rather common in real life. Such reliance on seeing all triplets in training data is problematic, as possible triplets in the wild are simply inexhaustible, which requires smarter models with better generalizability.

Motivation: Commonsense as a "coacher" for zero-shot relation prediction. Commonsense knowledge refers to general facts about the world that empower human beings to reason over unfamiliar scenarios. Motivated by this process from humans' perspective, in this work, we propose to integrate commonsense knowledge to alleviate the inexhaustible-triplet problem and improve the performance of zero-shot relation prediction in SGG.

The illustration of our problem and insights are illustrated in Figure 1. Specifically, the commonsense knowledge utilized in this paper comes from ConceptNet [13], a crowd-sourced semantic knowledge graph containing rich structured knowledge regarding real-world concepts.

Insight 1: Neighbor commonsense reflects semantic similarity. In ConceptNet, the neighbor similarity between two individual nodes indicates their semantic similarity in the real world. For example, in Figure 1, *child* and *man* share many common neighbors such as *fun*, *sleep*, *boy* and so on, which indicates that *child* and *man* may be similar and thus have similar interactions with other entities. If the model sees a triplet $\langle$*man, eating, pizza*$\rangle$ in the training data, then with the knowledge that *child* is semantically similar to *man*, it should more easily recognize triplets like $\langle$*child, eating, pizza*$\rangle$ from unseen but similar images. Therefore, we propose to leverage the semantic similarity between two detected entities by modeling their neighborhood overlap in ConceptNet.

Insight 2: Path commonsense reflects relation similarity. Nodes are connected by paths composed of multiple consecutive edges in ConceptNet. As is shown on the right in Figure 1, the entity pairs of $(child, pizza)$ and $(man, pizza)$ share common intermediate paths like $\langle$*RelatedTo, human, Desires, food, RelatedTo*$\rangle$. This similarity of intermediate paths indicates that the relations between *man* and *pizza* may be similar to those between *child* and *pizza*. If there is a triplet $\langle$*man, eating, pizza*$\rangle$ in the training data, then the model should tend to predict the relation *eating* given (*child*, *pizza*) in an unseen but similar image. Following the idea above, we propose to infer the relation between two entities by modeling their path coincidence with other entity pairs in ConceptNet.

Approach: Scene Graph Relation Prediction through Commonsense Knowledge Integration. In this work, we propose a novel framework that integrates external commonsense knowledge into SGG for relation prediction on zero-shot triplets, which we term as $\underline{\text{C}}$omm$\underline{\text{O}}$nsense-integr$\underline{\text{A}}$ted s$\underline{\text{C}}$ene grap$\underline{\text{H}}$ r$\underline{\text{E}}$lation p$\underline{\text{R}}$ediction, COACHER for brevity (Figure 3). To be specific, we investigate the utility of external commonsense knowledge through real data analysis. With the validated effectiveness of both neighbor- and path-based commonsense knowledge from ConceptNet, we design three modules for different levels of knowledge integration, which generate auxiliary knowledge embedding for generalizable relation prediction.

In summary, our main contributions are three-fold.

- We analyze the ignorance of zero-shot triplets by existing SGG models and validate the potential utility of commonsense knowledge from ConceptNet through real data analysis (Section 2).
- Based on the state-of-the-art SGG framework, we integrate external commonsense knowledge regarding ConceptNet neighbors and paths to improve relation prediction on zero-shot triplets (Section 3).
- Extensive quantitative experiments and qualitative analyses on the widely-used SGG benchmark dataset of Visual Genome demonstrate the effectiveness of our proposed COACHER framework. Particularly, COACHER achieves consistently better performance over state-of-the-art baselines on the original dataset, and outperforms them significantly on amplified datasets towards more severe zero-shot learning settings (Section 4).

2 Motivating Analysis

2.1 Ignorance yet Importance of Zero-Shot Triplets

In SGG, triplets are used to model entities with their relations. Among the great number of possible rational triplets in the wild, some of them exist in the training data while more others do not. The ability of correctly inferring zero-shot triplets can be extremely important to reflect the generalization capability of the model and its real utility in practice.

Although the performance of zero-shot scene graph generation was once studied in the early days on a small dataset [15], later researchers do not pay much attention to this setting. Until 2020, Tang et al. [21] first reported zero-shot performance on Visual Genome. However, they have not proposed any particular solutions to improve it.

Table 1. Performance (%) of three state-of-the-art models on non-zero-shot and zero-shot triplets on Visual Genome (Please refer to Section 4 for more details about the presented models).

Methods	**NM**			**NM+**			**TDE**		
MeanRecall@K	K=20	K=50	K=100	K=20	K=50	K=100	K=20	K=50	K=100
none-zero-shot	25.12	33.32	37.06	25.08	33.69	37.54	26.26	35.93	40.27
zero-shot	12.85	18.93	21.84	12.28	18.28	21.30	5.84	11.68	15.10

The ignorance of zero-shot settings causes existing methods a dramatic descent on the relation prediction on zero-shot triplets. Table 1 shows the performance of three state-of-the-art models on Visual Genome, the most widely used benchmark dataset for SGG. Note that *mean recall* is used here for performance evaluation, which is the average result of *triplet-wise* recall. Consistently, the mean recall of non-zero-shot triplets under different values of K can achieve almost twice of that on zero-shot ones, which demonstrates a concerning performance deterioration on zero-shot relation prediction.

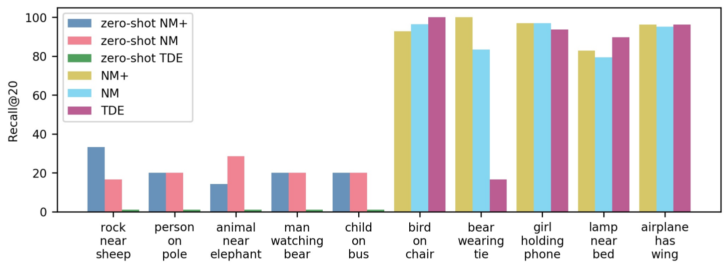


Fig. 2. Recall@20 performance on two different types of triplets. The five triplets on the left are zero-shot ones while the other five on the right are non-zero-shot.

Furthermore, although zero-shot triplets are not labeled in the training data, some of them are actually no less common in reality compared to the labeled ones. To give a more concrete illustration, we visualize the *recall@20* on five actual zero-shot triplets and another five non-zero-shot triplets in Figure 2. Consistent with the results in Table 1, performance of existing models on the five zero-shot triplets turns out to be much lower than that on the five non-zero-shot triplets. However, these zero-shot triplets represent very common relations such as ⟨*child, on, bus*⟩ which are in fact more common than some non-zero-shot ones such as ⟨*bear, wearing, tie*⟩. The performance on certain triplets like ⟨*bear, wearing, tie*⟩ is much better simply because they appear in the training data and got memorized by the model, but the utility of such memorization is rather limited in reality without the ability of generalization.

Lastly, due to the inexhaustibility of triplets in the real world, labeled datasets can only cover a limited portion of the extensive knowledge. Many common triplets simply do not appear in the dataset. For example, ⟨*men, holding, umbrella*⟩, ⟨*vehicle, parked on, beach*⟩,⟨*women, in, boat*⟩ and ⟨*fence, across, sidewalk*⟩, to name a few, are common triplets that never appear in Visual Genome, even though all involved objects and relations appear in other labeled triplets. Therefore, to propose a practical SGG model that can be widely used and assist various downstream tasks, we need to pay more attention to zero-shot triplets.

Motivated by the ignorance and importance of zero-shot triplets, in this work, we focus on integrating commonsense knowledge from external resources to improve the relation prediction performance on zero-shot triplets. Specifically, we leverage ConceptNet as the external knowledge resource from several other alternatives due to its wide coverage of concepts and accompanying semantic embeddings of concepts as useful features [13]. In ConceptNet, each concept (word or phrase) is modeled as a node and each edge represents the relation between two concepts. Thanks to its wide coverage, we are able to link each entity class in Visual Genome to one node in ConceptNet.

2.2 Commonsense Knowledge from ConceptNet Neighbors

Many entities in real life share similar semantic meanings, which can potentially support zero-shot relation prediction. Given an image from Visual Genome, we can detect multiple entities, where each entity belongs to a class. Among these entity classes, for example, *(girl, boy, woman, man, child)* are all human beings, so the model should learn to generalize human-oriented relations among them.

The semantic similarity among classes in Visual Genome can be viewed as the neighborhood similarity of their corresponding nodes in ConceptNet, which can be calculated with the Jaccard similarity of their neighbors:

$$J(\mathcal{V}_A, \mathcal{V}_B) = \frac{|\mathcal{N}_A \cap \mathcal{N}_B|}{|\mathcal{N}_A \cup \mathcal{N}_B|}, \tag{1}$$

where $\mathcal{V}_A$, $\mathcal{V}_B$ are nodes corresponding to two given classes, and $\mathcal{N}_A$, $\mathcal{N}_B$ are the neighbors of them in ConceptNet, respectively.

In order to validate the effectiveness of utilizing neighborhood similarity in ConceptNet as a measurement of semantic similarity in Visual Genome, we calculate the similarity between each pair of the top 150 mostly observed classes in Visual Genome, and rank their similarity in descending order. Results of the

Table 2. Top 20 pairs of similar entity classes.

1	*chair-seat*	**2**	*shoe-sock*	**3**	*hill-mountain*	**4**	*coat-jacket*	**5**	*house-building*
6	*airplane-plane*	**7**	*woman-girl*	**8**	*desk-table*	**9**	*window-door*	**10**	*men-man*
11	*shirt-jacket*	**12**	*house-room*	**13**	*room-building*	**14**	*girl-boy*	**15**	*plate-table*
16	*arm-leg*	**17**	*chair-table*	**18**	*tree-branch*	**19**	*cow-sheep*	**20**	*arm-hand*

top 20 most similar pairs are shown in Table 2. As can be seen, the top similar pairs such as $chair - seat$ indeed capture the semantic similarity between two classes, which validates the virtue of using ConceptNet neighbors to bring in commonsense knowledge for modeling the semantic similarity among entities.

2.3 Commonsense Knowledge from ConceptNet Paths

In ConceptNet, besides the one-hop information from neighbors, paths composed by multiple edges can further encode rich multi-hop information. Specifically, if two pairs of entities are connected by many same paths in ConceptNet, they are more likely to share similar relations. In order to investigate such path-relation correlation between node pairs on ConceptNet, we define MidPath as follows:

Definition 1 (MidPath). *Given two nodes $\mathcal{V}_A$ and $\mathcal{V}_B$ in the graph, a MidPath between $\mathcal{V}_A$ and $\mathcal{V}_B$ is defined as the sequence of all intermediate edges and nodes on a path from $\mathcal{V}_A$ to $\mathcal{V}_B$, excluding both the head and tail nodes.*

For example, given a path ⟨*people, RelatedTo, automobile, AtLocation, street*⟩ between nodes *people* and *street*, the corresponding MidPath is ⟨*RelatedTo, automobile, AtLocation*⟩.

Table 3. Top 3 related MidPaths for 10 relations.

Relation	Top1 MidPath	Top2 MidPath	Top3 MidPath
parked on	*RelatedTo-cars-RelatedTo*	*RelatedTo-driven-ReceivesAction*	*AtLocation-automobile-RelatedTo*
says	*RelatedTo*	*RelatedTo-communication_device-RelatedTo*	*RelatedTo-command-RelatedTo*
laying on	*RelatedTo-legs-RelatedTo*	*AtLocation*	*RelatedTo-human-RelatedTo*
wearing	*RelatedTo-body-RelatedTo*	*RelatedTo-dress-RelatedTo*	*RelatedTo-clothing-Desires*
against	*AtLocation-garage-AtLocation*	*RelatedTo-wall-RelatedTo*	*RelatedTo*
sitting on	*AtLocation*	*AtLocation-human-RelatedTo*	*RelatedTo-legs-RelatedTo*
walking in	*RelatedTo-home-RelatedTo*	*UsedFor-children-RelatedTo*	*RelatedTo-crowd-RelatedTo*
growing on	*RelatedTo-growth-RelatedTo*	*RelatedTo-leaves-RelatedTo*	*RelatedTo-stem-RelatedTo*
watching	*RelatedTo-human-RelatedTo*	*RelatedTo-date-RelatedTo*	*RelatedTo-female-RelatedTo*
playing	*RelatedTo-human-RelatedTo*	*RelatedTo*	*RelatedTo-clown-RelatedTo*

For each relation in Visual Genome, probability analysis is done to investigate the related MidPaths. The smallest unit in the dataset is a triplet ⟨*subject, relation, object*⟩. With nodes *subject* and *object*, we can extract a set of MidPaths $\mathcal{P}$ connecting them from ConceptNet. For each path $p_i \in \mathcal{P}$ and relation $r_j \in \mathcal{R}$, the number of co-occurrence $I(\mathcal{MP} = p_i, \mathcal{R} = r_j)$ can be counted. Following the formula below, we calculate the conditional probability of observing

MidPath p_i given a specific relation r_j:

$$P(\mathcal{MP} = p_i | \mathcal{R} = r_j) = \frac{I(\mathcal{MP} = p_i, \mathcal{R} = r_j)}{\sum_{p_k \in \mathcal{P}} I(\mathcal{MP} = p_k, \mathcal{R} = r_j)}. \quad (2)$$

To eliminate random effects, the probability of observing a random MidPath p_i is also calculated, as follows

$$P(\mathcal{MP} = p_i) = \frac{I(\mathcal{MP} = p_i)}{\sum_{p_k \in \mathcal{P}} I(\mathcal{MP} = p_k)}, \quad (3)$$

where $I(\mathcal{MP} = p_i)$ is the occurrence number of MidPath p_i. Now we can measure how significant MidPath p_i is given a specific relation r_j by

$$Score(p_i, r_j) = P(\mathcal{MP} = p_i | \mathcal{R} = r_j) - P(\mathcal{MP} = p_i). \quad (4)$$

The higher the score is, the more significantly MidPath p_i can imply relation r_j. Table 3 shows top three MidPaths with highest scores for 10 relations. Clearly, these top MidPaths are semantically meaningful and potentially beneficial for generalizing relation predictions among entity pairs.

3 COACHER

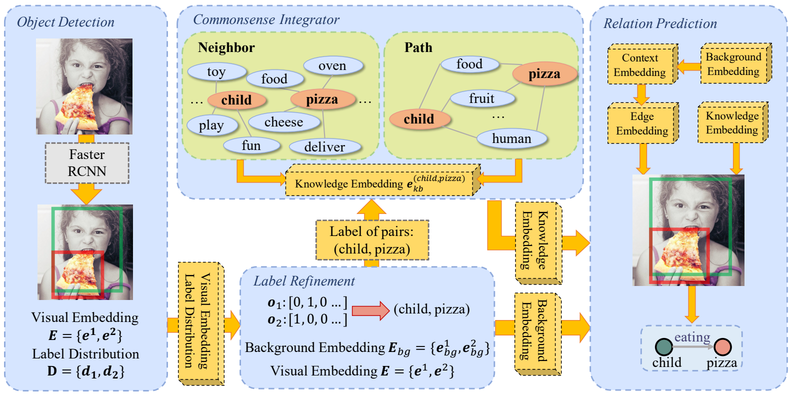


Fig. 3. The overall framework of our proposed COACHER.

In this section, we present the proposed SGG with commonsense knowledge integration model COACHER in detail. Figure 3 shows the major components in COACHER: (a) object detection and refinement, (b) commonsense knowledge embedding generation, and (c) commonsense enhanced relation prediction.

3.1 Backbone Scene Graph Generation Pipeline

Scene graph generation is a task aiming to understand visual scenes by extracting entities from images and predicting the semantic relations between them. Various technique has been explored for scene graph generation (Section 5). Here we adopt one of the state-of-the-art pipelines from Neural Motif [34] as the backbone framework. We mainly focus on commonsense-integrated relation prediction for zero-shot triplets, which is crucial for practice as discussed in Section 2.

The backbone scene graph generation pipeline includes three components:

Step1: object detection. With the development of deep learning, some popular frameworks such as R-CNN [3, 19], YOLO [18] and SSD [14] achieve impressive performance on this task. Following previous literature [5,21], we adopt a pre-trained Faster R-CNN model as the detector in our framework. In this step, with the input of a single image $\mathcal{I}$, the output includes: a set of region proposals $B = \{b_1, \cdots, b_n\}$, a set of distribution vectors $\boldsymbol{D} = \{\boldsymbol{d}_1, \cdots, \boldsymbol{d}_n\}$, where $\boldsymbol{d}_i \in \mathbb{R}^{|\mathcal{C}|}$ is a label probabilities distribution and $|\mathcal{C}|$ is the number of classes, as well as a visual embedding $\boldsymbol{E} = \{\boldsymbol{e}^1, \cdots, \boldsymbol{e}^n\}$ for each detected object.

Step2: label refinement. Based on the label distribution $\boldsymbol{D}$ generated from Step1, we conduct label refinement to generate a one-hot vector of entity classes for each region proposal, which will be used for relation prediction.

First, background embeddings $\boldsymbol{E}_{bg}$ containing information from both region proposal level and global level of the image are generated using a bi-LSTM:

$$\boldsymbol{E}_{bg} = \text{biLSTM}([\boldsymbol{e}^i; \text{MLP}(\boldsymbol{d}_i)]_{i=1,\cdots,n}), \tag{5}$$

where $\boldsymbol{E}_{bg} = [\boldsymbol{e}^1_{bg}, \boldsymbol{e}^2_{bg}, \cdots, \boldsymbol{e}^n_{bg}]$ is the hidden state of the last layer in LSTM and n is the number of region proposals. Then, an LSTM is used to decode each region proposal embedding $\boldsymbol{e}^i_{bg}$:

$$\boldsymbol{h}_i = \text{LSTM}([\boldsymbol{e}^i_{bg}; \boldsymbol{o}_{i-1}]), \tag{6}$$

$$\boldsymbol{o}_i = \text{argmax}(\text{MLP}(\boldsymbol{h}_i)). \tag{7}$$

$\boldsymbol{o}_i$ is the one-hot vector representing the refined class label of a region proposal.

Step3: relation prediction. After obtaining refined object labels for all region proposals, we use them to further generate context embeddings $\boldsymbol{E}_{ct}$:

$$\boldsymbol{E}_{ct} = \text{biLSTM}([\boldsymbol{e}^i_{bg}; \text{MLP}(\boldsymbol{o}_i)]_{i=1,\cdots,n}), \tag{8}$$

where $\boldsymbol{E}_{ct} = [\boldsymbol{e}^1_{ct}, \boldsymbol{e}^2_{ct}, \cdots, \boldsymbol{e}^n_{ct}]$, which are then used to extract edge embeddings $\boldsymbol{e}_{eg}$ and predict the relation between each pair of bounding boxes:

$$\boldsymbol{e}^{(i,j)}_{eg} = \text{MLP}(\boldsymbol{e}^i_{ct}) \circ \text{MLP}(\boldsymbol{e}^j_{ct}) \circ (\boldsymbol{e}^i \cup \boldsymbol{e}^j), \tag{9}$$

$$r_{(i,j)} = \text{argmax}(\text{MLP}([\boldsymbol{e}^{(i,j)}_{eg}; \boldsymbol{e}^{(i,j)}_{kb}])), \tag{10}$$

where $\circ$ represents element-wise product, $\boldsymbol{e}^{(i,j)}_{kb}$ is the commonsense knowledge embedding obtained from ConceptNet, which we will introduce next.

3.2 Commonsense Integrator

Commonsense knowledge integration is achieved by computing $\boldsymbol{e}_{kb}$ from external resources. Specifically, we use ConceptNet [13] here as the source of external commonsense knowledge. ConceptNet is a knowledge graph that connects words and phrases of natural language with labeled edges. It is constructed from rich resources such as Wiktionary and WordNet. With the combination of these resources, ConceptNet contains over 21 million edges and over 8 million nodes, covering all of the entity classes in Visual Genome. Besides, it also provides a semantic embedding for each node, which can serve as a semantic feature. Here we develop three types of integrators to generate commonsense knowledge embeddings from ConceptNet.

Neighbor integrator. ConceptNet is a massive graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are the node set and edge set, respectively. Each detected entity can be seen as a node in ConceptNet. Given a class $c \in \mathcal{C}$ of a detected entity, its neighborhood information can be collected as follows:

$$c \xrightarrow{\text{link}} \mathcal{V}_c \xrightarrow{\text{retrieve neighbors}} \mathcal{N}_c = \{\mathcal{V}_n | (\mathcal{V}_c, \mathcal{V}_n) \in \mathcal{E}\}. \tag{11}$$

Denote $\boldsymbol{F} \subset \mathbb{R}^{|\mathcal{V}| \times k}$ as the feature matrix of all nodes in ConceptNet from [20], where k is the dimension of the feature vector. Neighbor embedding $\boldsymbol{e}_{nb}^{c}$ of node $\mathcal{V}_c$ is calculated as the average over all of its neighbors' embeddings:

$$\boldsymbol{e}_{nb}^{c} = \frac{1}{|\mathcal{N}_c|} \sum_{\mathcal{V}_n \in \mathcal{N}_c} \boldsymbol{F}_n, \tag{12}$$

where $\boldsymbol{F}_n$ is the n_{th} row of $\boldsymbol{F}$. For relation prediction, given a pair of detected entities with classes (a, b), the neighbor-based commonsense knowledge embedding $\boldsymbol{e}_{kb}^{(a,b)}$ of this pair is calculated as:

$$\boldsymbol{e}_{kb}^{(a,b)} = \text{ReLU}(\text{MLP}([\boldsymbol{e}_{nb}^{a}; \boldsymbol{e}_{nb}^{b}])), \tag{13}$$

where MLP denotes the multi-layer perceptron.

Path integrator. Given a pair of entities with classes (a, b) recognized from the object detection model, a set of paths connecting them can be obtained following the procedure below:

$$(a, b) \xrightarrow{\text{link}} (\mathcal{V}_a, \mathcal{V}_b) \xrightarrow{\text{retrieve paths}} \mathcal{P}_{(a,b)} = \{p | p = \{\mathcal{V}_a, \mathcal{V}_1, \cdots, \mathcal{V}_b\}\}. \tag{14}$$

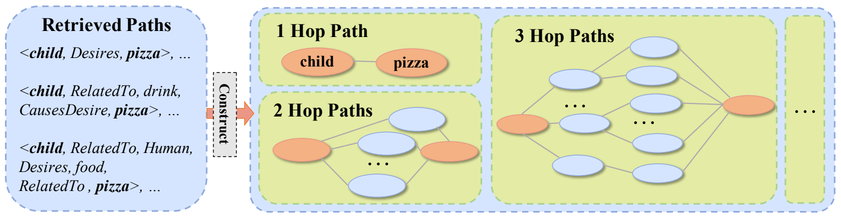

Retrieved Paths
<child, Desires, pizza>, ...
<child, RelatedTo, drink, CausesDesire, pizza>, ...
<child, RelatedTo, Human, Desires, food, RelatedTo , pizza>, ...
Construct
1 Hop Path
child
pizza
2 Hop Paths
3 Hop Paths

Fig. 4. Graphic representation construction from retrieved paths.

We classify these extracted paths based on their number of hops. The graphical representation of paths is shown in Figure 4. Each set of *l-Hop Paths* naturally constitutes a small graph $G_l^{(a,b)}$, where l is the number of hops on the paths. The goal is to learn a representation of all graphs $\{G_l^{(a,b)}\}_{l=1}^{L}$, where L is often a small number like 2 or 3, since longer paths are too noisy and hard to retrieve.

Classical sequence models such as LSTM cannot handle very short paths effectively. Inspired by message passing network for graph representation learning [2,16], we design a neural message passing mechanism to learn a representation for each set of *l-Hop Paths*, and then combine them as the final path-based commonsense knowledge embedding for pair (a, b).

Specifically, we design the message passing mechanism as follows:

$$\text{MSG}_v^t(\mathcal{V}_a) = \text{MSG}_v^{t-1}(\mathcal{V}_a) + \sum_{v \in \mathcal{N}_a} \text{MSG}_e^t(v), \tag{15}$$

$$\text{MSG}_e^t(v) = \text{MLP}(\text{MSG}_v^{t-1}(v)), \tag{16}$$

where $\mathrm{MSG}_v^0(v)$ is initialized as $\boldsymbol{F}_v$, *i.e.*, the original node features from [20].

In order to update the embedding of each node with all hops on the corresponding paths, given $G_l^{(a,b)}$, we iterate the above process $t = l$ times to obtain the final node embeddings $\boldsymbol{T}_l^{(a,b)}$ on each set of *l-Hop Paths* in $G_l^{(a,b)}$.

To join the information of multiple paths on each graph, we select and aggregate the most important paths. As a simplification, we adopt the GlobalSortPool operator [35] on the embeddings of all nodes in $G_l^{(a,b)}$ as follows:

$$\boldsymbol{e}_g^{(a,b),l} = \mathrm{GlobalSortPool}(\boldsymbol{T}_l^{(a,b)}), \tag{17}$$

which learns to sort all nodes by a specific embedding channel and pool the top K (*e.g.*, K=5) nodes on all embedding channels.

Finally, we aggregate the path embeddings $\{\boldsymbol{e}_g^{(a,b),l}\}_{l=1}^{L}$ of different length l through vector concatenation.

Fused integrator. To fuse neighbor- and path-based commonsense knowledge, we inject the neighbor-based knowledge into the path-based knowledge by initializing $\mathrm{MSG}_v^0(v)$ in Eq. 15 as the element-wise mean between $\boldsymbol{e}_{nb}$ from Eq. 12 and the original node features $\boldsymbol{F}$ as follows:

$$\mathrm{MSG}_v^0(v) = \mathrm{MEAN}(\boldsymbol{F}_v, \boldsymbol{e}_{nb}^v), \tag{18}$$

where v can be replaced as a and b for the entity class pair (a, b).

4 Experiments

4.1 Experimental Settings

Original whole dataset. For scene graph generation, we use the Visual Genome dataset [8], a commonly used benchmark for SGG, to train and test our framework. This dataset contains 108,077 images, where the number of classes and relations are 75,729 and 40,480, respectively. However, 92% of relationships have no more than 10 instances. Therefore, we follow the widely used split strategy on Visual Genome [5, 21, 24] that selects the most frequent 150 object classes and 50 relations as a representative. Besides, we use 70% images as well as their corresponding entities and relations as the training set, and the other 30% of images are left out for testing. A 5k validation set is split from the training set for parameter tuning.

Zero-shot amplified dataset. To further investigate the model's generalization ability in more severe zero-shot settings, we reduce the information that the model can leverage during training by constructing another zero-shot amplified dataset. This is achieved by simply removing images containing less common relations from the training data. As a result, the triplet numbers of the last thirty common relations are halved, while the triplet numbers of the first twenty common relations mostly remain the same. In this way, we exacerbate the difficulty for the model, especially on predicting relations for zero-shot triplets.

Compared algorithms. We compare COACHER with four baseline methods.

- **NeuralMotifs (NM)** [34] is a strong baseline which is widely-compared on Visual Genome for SGG.
- **NeuralMotifs with Knowledge Refinement Network (NM+)** [5] is the only existing method that leverages external knowledge for SGG, which is the closest to ours. This method mainly contains two new parts, knowledge refinement and image reconstruction. We add its knowledge refinement part on top of Neural Motifs, which we call NM+.
- **TDE** [21] is the current state-of-the-art method for scene graph generation. This work is also the first one reporting zero-shot performance on Visual Genome but it does not take effort to improve it.
- **CSK-N** is a baseline based on our framework which makes predictions without visual information. Given a pair of entities, we predict their relation using only the neighbor-based commonsense knowledge embedding.

Evaluation metrics. Since the purpose of this work is to improve performance on zero-shot triplets, we follow the relation classification setting for evaluation [21]. Specifically, we use the following two metrics and focus on their evaluations on the zero-shot subset of the whole testing data:

- **zR@K**(zero-shot Recall@K): Recall@K is the earliest and the most widely accepted metric in SGG, first used by Lu et al. [15]. Since the relation between a pair of entities is not complete, treating it as a retrieval problem is more proper than a classification problem. Here we take the Recall@K on zero-shot subset and shorten it as zR@K.
- **ng-zR@K**(zero-shot no-graph-constraint Recall@K): No-graph-constraint Recall@K is first used by Newell et al. for SGG [17]. It allows a pair of entities to have multiple relations, which significantly improves the recall value. Here we take the ng-R@K on zero-shot subset and shorten it as ng-zR@K.

4.2 Hyper-parameter setting

A pre-trained and frozen Faster-RCNN [19] equipped with the ResNeXt-101-FPN [6] backbone is used as the object detector for all models. Batch size and the max iteration number are set as 12 and 50,000 respectively. The learning rate begins with 1.2×10^{-1} with a decay rate of 10 and a stepped scheduler based on the performance stability on the validation set. A Quadro RTX 8000 GPU with 48GB of memory is used for our model training. For path length, here we only use 1 and 2 hop paths ($L = 2$) due to the GPU memory limitation. Based on observation from humans' perspective, these short paths are also the more informative ones compared with longer paths.

4.3 Performance evaluations

Table 4 and Table 5 show performance of three variants of COACHER with different knowledge integrators (COACHER-N, COACHER-P, COACHER-N+P)

Table 4. Zero-shot performance (%) on the original whole dataset.

Method	zR@20	zR@50	zR@100	ng-zR@20	ng-zR@50	ng-zR@100
NM	13.05 ± 0.06	19.03 ± 0.22	21.98 ± 0.22	15.16 ± 0.49	28.78 ± 0.57	41.52 ± 0.79
NM+	12.35 ± 0.28	18.10 ± 0.13	21.13 ± 0.24	14.47 ± 0.11	27.93 ± 0.15	40.84 ± 0.28
TDE	8.36 ± 0.25	14.35 ± 0.27	18.04 ± 0.46	9.84 ± 0.33	19.28 ± 0.56	28.99 ± 0.44
CSK-N	5.95 ± 0.62	10.12 ± 0.79	13.05 ± 0.64	8.15 ± 0.57	16.79 ± 0.62	26.19 ± 1.19
COACHER-N	12.73 ± 0.22	18.88 ± 0.12	21.88 ± 0.11	15.10 ± 0.47	28.73 ± 0.21	41.06 ± 0.26
COACHER-P	12.24 ± 0.17	18.12 ± 0.16	21.55 ± 0.39	14.39 ± 0.46	28.90 ± 0.43	40.98 ± 0.45
COACHER-N+P	13.42 ± 0.28	19.31 ± 0.27	22.22 ± 0.29	15.54 ± 0.27	29.31 ± 0.27	41.39 ± 0.22

Table 5. Zero-shot performance (%) on the zero-shot amplified dataset.

Method	zR@20	zR@50	zR@100	ng-zR@20	ng-zR@50	ng-zR@100
NM	11.98 ± 0.09	17.86 ± 0.13	20.48 ± 0.02	13.98 ± 0.33	27.43 ± 0.72	39.33 ± 0.77
NM+	11.82 ± 0.06	17.27 ± 0.34	20.10 ± 0.31	13.83 ± 0.18	26.92 ± 0.25	38.71 ± 0.32
TDE	5.67 ± 0.03	11.08 ± 0.40	14.20 ± 0.22	6.51 ± 0.05	18.61 ± 0.05	32.20 ± 0.47
CSK-N	5.65 ± 0.34	9.55 ± 0.40	11.74 ± 0.98	7.35 ± 0.45	15.04 ± 0.85	23.91 ± 1.05
COACHER-N	11.79 ± 0.70	17.42 ± 0.88	20.08 ± 1.11	14.14 ± 0.78	26.74 ± 1.38	38.44 ± 1.52
COACHER-P	12.17 ± 0.84	18.02 ± 1.23	20.58 ± 1.52	14.53 ± 0.92	27.66 ± 1.21	38.86 ± 1.48
COACHER-N+P	12.29 ± 0.17	17.85 ± 0.33	20.26 ± 0.46	14.71 ± 0.58	27.67 ± 0.82	39.57 ± 0.54

as well as four baselines on the original and amplified datasets, respectively. The results from baseline methods are reported under their best settings. Using a Bayesian correlated t-Test [1], there is 98% probability that COACHER-N+P is better than baseline methods. It is shown that on both datasets, our methods COACHER-N+P and COACHER-P surpass all baselines and achieve by far the highest results on both zR@K and ng-zR@K. The performance gains in Table 5 are more significant than Table 4. Such observations directly support the effectiveness of COACHER in zero-shot relation prediction, indicating its superior generalizability as we advocate in this work. Note that, consistent with their own report and our analysis in Table 1, TDE reaches state-of-the-art on non-zero-shot triplets, but performs rather poor on zero-shot ones.

Further amplifications on testing data. In order to observe how powerful our model is on harder zero-shot triplets, we further manipulate the testing data by removing zero-shot triplets whose relations are more commonly observed in other triplets in the training data. As is shown in Figure 5, our proposed method COACHER-N+P shows the least drop compared to other methods as triplets of the top 5 common relations are removed one-by-one from the testing data, which again indicates the advantageous generalization ability of our model.

4.4 Case studies

In this subsection, we illustrate the contributions of both neighbor- and path-based commonsense knowledge integrators for SGG with real zero-shot relation prediction examples, shown in Figure 6.

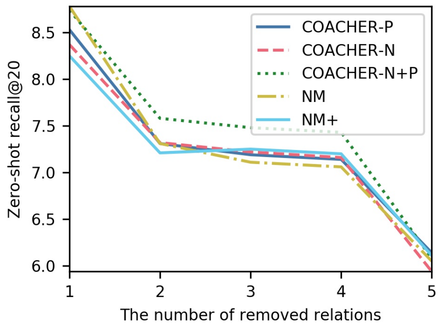
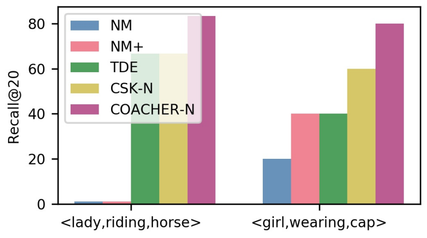
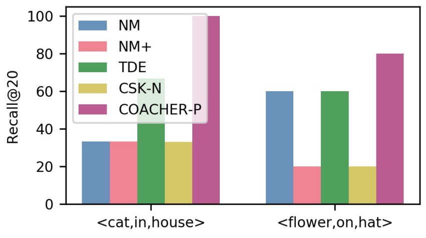


Fig. 5. Performance on manipulated testing data (we removed TDE and CSK-N here due to their rather poor performance on zR@20).



(a) neighbor-based commonsense helps



(b) path-based commonsense helps

Fig. 6. Case studies on integration of neighbor- and path-based commonsense.

Neighbor. The contribution of neighbor-based commonsense mainly comes from the semantic similarity implied by the neighborhood overlap between entities. As is shown in Figure 6a, although triplet ⟨*lady, riding, horse*⟩ has never appeared in the training data, several similar triplets such as ⟨*man, riding, horse*⟩ and ⟨*person, riding, horse*⟩ are observed in the training data, where their occurrences reach 194 times and 83 times respectively. Estimated from the nodes neighborhood similarity in ConceptNet, the semantic meaning between *lady*, *man* and *person* are similar to each other. COACHER can leverage this external knowledge to improve prediction performance on zero-shot triplets. Similarly, in another case of ⟨*girl, wearing, cap*⟩, where *cap* is semantically similar with *hat*, since ⟨*girl, wearing, hat*⟩ is commonly observed in the training data, COACHER can use the neighbor-based commonsense knowledge to make better relation predictions on zero-shot triplets. In contrast, other baseline methods fail in both cases because the generalizable semantic knowledge is hard to directly learn from limited visual information.

Path. The contribution of path-based commonsense mainly comes from the relations implied by *MidPaths* between entities. As shown in Figure 6b, for the triplet ⟨*flower, on, hat*⟩, we can find an inductive path ⟨*flower, RelatedTo, decoration, UsedFor, hat*⟩, which promotes the prediction of relation *on*. Similarly, given the pair (*cat*, *house*), multiple paths like ⟨*cat, AtLocation, home, RelatedTo, house*⟩,

⟨*cat, RelatedTo, house*⟩ and ⟨*cat, AtLocation, apartment, Antonym, house*⟩ can be found. All of them are inductive towards the correct relation of *in*.

5 Related Work

5.1 Scene Graph Generation

Scene graph generation (SGG) has been widely investigated over the last decade due to its potential benefit for various visual reasoning tasks. Lu et al. [15] train visual models for entities (e.g. "man" and "bicycle") and predicates ("riding" and "pushing") individually, and then combine them for the prediction of multiple relations. Xu et al. [24] propose a joint inference model to iteratively improve predictions on entities and relations via message passing. Liao [30] further integrate physical constraints between entities to extract support relations. Importantly, Zellers et al. [34] raise the bias problem into attention that entity labels are highly predictive of relation labels and give a strong baseline. Recently, Tang et al. [21] present a general framework for unbiased SGG based on causal inference, which performs as the current state-of-the-art SGG method.

All methods above do not leverage external knowledge. The only exception is Gu et al. [5], which adds a knowledge refinement module into the SGG pipeline to leverage external knowledge. It is indeed the closest work to ours and compared as a major baseline. However, they do not consider the zero-shot relation prediction problem and their integration of external knowledge is rather coarse and limited compared with ours.

5.2 External Knowledge Enhanced Deep Neural Networks

Knowledge Bases (KBs) can be used as an external knowledge to improve various down-stream tasks. The natural language processing and computer vision communities have proposed several ways to benefit deep neural networks from KBs. The most straightforward and efficient way is to represent external knowledge as an embedding, and then combine it with other features to improve model's performance. For example, in order to incorporate external knowledge to answer open-domain visual questions with dynamic memory networks, Li et al. [12, 23] extract the most informative knowledge and feed them into the neural network after embedding the candidate knowledge into a continuous feature space to enhance QA task. The graph community also utilizes external knowledge to enhance graph learning [25–29]. Besides, external knowledge can be added to loss function or perform as a regularization in the training process. For example, Yu et al. [33] obtain linguistic knowledge by mining from internal training annotations as well as external knowledge from publicly available text.

However, external knowledge has hardly been utilized in SGG, mainly because the direct entity-level embeddings can hardly help in object detection, whereas what kind of embeddings are helpful in what kind of relation prediction has remained unknown before our exploration.

6 Conclusion

Scene graph generation has been intensively studied recently due to its potential benefit in various downstream visual tasks. In this work, we focus on the key challenge of SGG, *i.e.*, relation prediction on zero-shot triplets. Inspired by the natural ability of human beings to predict zero-shot relations from learned commonsense knowledge, we design integrators to leverage neighbor- and path-based commonsense from ConceptNet. We demonstrate the effectiveness of our proposed COACHER through extensive quantitative and qualitative experiments on the most widely used benchmark dataset of Visual Genome.

For future works, more in-depth experiments can be done to study the external knowledge graphs for SGG. Although the current ConceptNet is comprehensive enough to cover all entities detected from Visual Genome images, the relations it models are more from the factual perspectives, such as *has property*, *synonym*, *part of*, whereas the relations in scene graphs are more from the actional perspectives, such as the spatial or dynamical interactions among entities. One promising direction based on this study is to construct a scene-oriented commonsense knowledge graph specifically for visual tasks, while the downstream training process can further refine the graph construction. In this way, the gap between these two isolated communities, *i.e.*, visual reasoning and knowledge extraction, can be bridged and potentially enhance each other.

References

1. Benavoli, A., Corani, G., Demšar, J., Zaffalon, M.: Time for a change: A tutorial for comparing multiple classifiers through bayesian analysis. JMLR **18**, 2653–2688 (2017)
2. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: ICML (2017)
3. Girshick, R.: Fast r-cnn. In: ICCV (2015)
4. Gu, J., Cai, J., Wang, G., Chen, T.: Stack-captioning: Coarse-to-fine learning for image captioning. In: AAAI (2018)
5. Gu, J., Zhao, H., Lin, Z., Li, S., Cai, J., Ling, M.: Scene graph generation with external knowledge and image reconstruction. In: CVPR (2019)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
7. Hudson, D.A., Manning, C.D.: Gqa: A new dataset for real-world visual reasoning and compositional question answering. In: CVPR (2019)
8. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. IJCV **123**, 32–73 (2017)
9. Li, B., Qi, X., Lukasiewicz, T., Torr, P.: Controllable text-to-image generation. In: NeurIPS (2019)
10. Li, B., Qi, X., Torr, P., Lukasiewicz, T.: Lightweight generative adversarial networks for text-guided image manipulation. In: NeurIPS (2020)
11. Li, B., Qi, X., Torr, P., Lukasiewicz, T.: ManiGAN: text-guided image manipulation. In: CVPR (2020)

12. Li, G., Su, H., Zhu, W.: Incorporating external knowledge to answer open-domain visual questions with dynamic memory networks (2017)
13. Liu, H., Singh, P.: Conceptnet — a practical commonsense reasoning tool-kit. BT Technology Journal **22**, 211–226 (2004)
14. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: ECCV (2016)
15. Lu, C., Krishna, R., Bernstein, M., Fei-Fei, L.: Visual relationship detection with language priors. In: ECCV (2016)
16. Murphy, K., Weiss, Y., Jordan, M.I.: Loopy belief propagation for approximate inference: An empirical study. In: UAI (1999)
17. Newell, A., Deng, J.: Pixels to graphs by associative embedding. In: NeurIPS (2017)
18. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv (2018)
19. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NeurIPS (2015)
20. Speer, R., Chin, J., Havasi, C.: Conceptnet 5.5: An open multilingual graph of general knowledge. In: AAAI (2017)
21. Tang, K., Niu, Y., Huang, J., Shi, J., Zhang, H.: Unbiased scene graph generation from biased training. CVPR (2020)
22. Teney, D., Liu, L., van Den Hengel, A.: Graph-structured representations for visual question answering. In: CVPR (2017)
23. Wu, Q., Shen, C., Wang, P., Dick, A., van den Hengel, A.: Image captioning and visual question answering based on attributes and external knowledge. TPAMI **40**(6) (2018)
24. Xu, D., Zhu, Y., Choy, C.B., Fei-Fei, L.: Scene graph generation by iterative message passing. In: CVPR (2017)
25. Yang, C.: Multi-facet graph mining with contextualized projections. PhD Thesis (2020)
26. Yang, C., Xiao, Y., Zhang, Y., Sun, Y., Han, J.: Heterogeneous network representation learning: a unified framework with survey and benchmark. TKDE (2020)
27. Yang, C., Pal, A., Zhai, A., Pancha, N., Han, J., and Rosenberg, C. and Leskovec, J.: Multisage: empowering graphsage with contextualized multi-embedding on web-scale multipartite networks. In: KDD (2020)
28. Yang, C., Zhang, J., Han, J.: Co-embedding network nodes and hierarchical labels with taxonomy based generative adversarial networks. In: ICDM (2020)
29. Yang, C., Zhuang, P., Shi, W., Luu, A., Pan, L.: Conditional structure generation through graph variational generative adversarial nets. In: NeurIPS (2019)
30. Yang, M.Y., Liao, W., Ackermann, H., Rosenhahn, B.: On support relations and semantic scene graphs. ISPRS Journal of Photogrammetry and Remote Sensing **131**, 15–25 (2017)
31. Yang, X., Tang, K., Zhang, H., Cai, J.: Auto-encoding scene graphs for image captioning. In: CVPR (2019)
32. Yao, T., Pan, Y., Li, Y., Mei, T.: Exploring visual relationship for image captioning. In: ECCV (2018)
33. Yu, R., Li, A., Morariu, V.I., Davis, L.S.: Visual relationship detection with internal and external linguistic knowledge distillation. In: ICCV (2017)
34. Zellers, R., Yatskar, M., Thomson, S., Choi, Y.: Neural motifs: Scene graph parsing with global context. CVPR (2018)
35. Zhang, M., Cui, Z., Neumann, M., Chen, Y.: An end-to-end deep learning architecture for graph classification. In: AAAI (2018)
36. Zhu, Y., Zhang, C., Ré, C., Fei-Fei, L.: Building a large-scale multimodal knowledge base system for answering visual queries. arXiv preprint arXiv:1507.05670 (2015)