Unpaired Image Captioning via Scene Graph Alignments

Jiuxiang Gu[1], Shafiq Joty[1,4], Jianfei Cai[1,2], Handong Zhao[3], Xu Yang[1], Gang Wang[5]
[1]Nanyang Technological University, Singapore [2]Monash University, Australia
[3]Adobe Research, USA [4]Salesforce Research Asia, Singapore [5]Alibaba Group, China
{jgu004, srjoty, asjfcai, s170018}@ntu.edu.sg, hazhao@adobe.com, gangwang6@gmail.com

Abstract

Most of current image captioning models heavily rely on paired image-caption datasets. However, getting large scale image-caption paired data is labor-intensive and time-consuming. In this paper, we present a scene graph-based approach for unpaired image captioning. Our framework comprises an image scene graph generator, a sentence scene graph generator, a scene graph encoder, and a sentence decoder. Specifically, we first train the scene graph encoder and the sentence decoder on the text modality. To align the scene graphs between images and sentences, we propose an unsupervised feature alignment method that maps the scene graph features from the image to the sentence modality. Experimental results show that our proposed model can generate quite promising results without using any image-caption training pairs, outperforming existing methods by a wide margin.

1. Introduction

Today's image captioning models heavily depend on paired image-caption datasets. Most of them employ an encoder-decoder framework [33, 9, 13, 12, 37], which uses a convolutional neural network (CNN) [16] to encode an image into a feature vector and then a recurrent neural network (RNN) to decode it into a text sequence. However, it is worthwhile noticing that the overwhelming majority of image captioning studies are conducted in English [1]. The bottleneck is the lack of large scale image-caption paired datasets in other languages, and getting such paired data for each target language requires human expertise in a time-consuming and labor-intensive process.

Several encoder-decoder models have been proposed in recent years for unsupervised neural machine translation [23, 4]. The key idea of these methods mainly relies on training denoising auto-encoders for language modeling and on sharing latent representations across the source and target languages for the encoder and the decoder. Despite the promising results achieved by the unsupervised neural machine translation, unpaired image-to-sentence translation is far from mature.

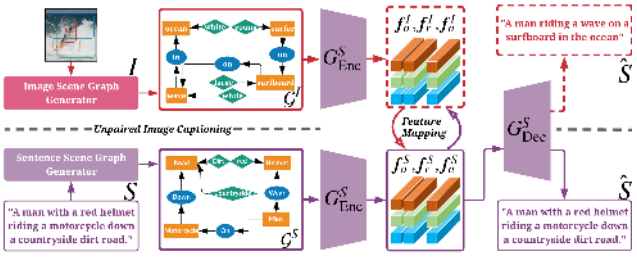


Figure 1: Illustration of our graph-based learning method. Our model consists of one visual scene graph detector (Top-Left), one fixed off-the-shelf scene graph language parser (Bottom-Left), a scene graph encoder G^S_{Enc}, a sentence decoder G^S_{Dec}, and a feature mapping module.

Recently, there have been few attempts at relaxing the requirement of paired image-caption data for this task. The first work in this direction is the pivot-based semi-supervised solution proposed by Gu *et al*. [14], where they take a pivot language as a bridge to connect the source image and the target language caption. Their method requires an image-text paired data for the pivot language (Chinese), and a parallel corpus for the pivot to target translation. Feng *et al*. [10] move a step further, where they conduct purely unsupervised image captioning without relying on any labeled image-caption pairs. Their method uses a sentence discriminator along with a visual concept detector to connect the image and the text modalities through adversarial training. Although promising, the results of the existing methods are still far below compared to their paired counterparts.

Unlike unsupervised neural machine translation where the encoders can be shared across the source and target languages, due to the different structures and characteristics of image and text modalities, the encoders of image and sentence cannot be shared to connect the two modalities. The critical challenge in unpaired image captioning is, therefore, the gap of information misalignment in images and sentences, so as to fit the encoder-decoder framework.

Fortunately, with recent breakthroughs in deep learning and image recognition, higher-level visual understanding tasks such as scene graph construction have become popular research topics with significant advancements [41, 6, 8, 7, 40, 35, 17]. Scene graph, as an abstraction of objects and their complex relationships, provide rich semantic information of an image. The value of scene graph representation has been proven in a wide range of vision-language tasks, such as visual question answering [31] and paired image captioning [35].

Considering the significant challenges that unpaired image captioning problem poses in terms of different characteristics between visual and textual modalities, in this paper, we propose a scene graph-based method that exploits the rich semantic information captured by scene graphs. Our framework comprises an image scene graph generator, a sentence scene graph generator, a scene graph encoder, a sentence decoder, and a feature alignment module that maps the features from image to sentence modality. Figure 1 sketches our solution. We first extract the sentence scene graphs from the sentence corpus and train the scene graph encoder and the sentence decoder on the text modality. To align the scene graphs between images and sentences, we use CycleGAN [42] to build the data correspondence between the two modalities. Specifically, given the unrelated image and sentence scene graphs, we first encode them with the scene graph encoder trained on the sentence corpus. Then, we perform unsupervised cross-modal mapping for feature level alignments with CycleGAN. By mapping the features, the encoded image scene graph is pushed close to the sentence modality, which is then used effectively as input to the sentence decoder to generate meaningful sentences.

The main contributions of this work include: (1) a novel scene graph-based framework for unpaired image captioning; (2) an unsupervised feature alignment method that learns the cross-modal mapping without any paired data. Our experimental results demonstrate the effectiveness of our proposed model in producing quite promising image captions. The comparison with recent unpaired image captioning methods validates the superiority of our method.

2. Background

Paired Image Captioning. Image captioning has been extensively studied in the past few years. Most of the existing approaches are under the paired setting, that is, input images come with their corresponding ground-truth captions [33, 19, 38, 13, 12, 36]. One classic work in this setting is [33], in which an image is encoded with a CNN, and the sentence is decoded with a Long Short-Term Memory (LSTM) network. Following this, many methods have been proposed to improve this encoder-decoder method. One of the notable improvements is the attention mechanism [34, 13, 3], which allows the sentence decoder to dynamically focus on some related image regions during the caption generation process. Some other works explore other architectures for language modeling [15, 30]. For example, Gu *et al*. [15] introduce a CNN-based language model for image captioning. Another theme of improvements is to use reinforcement learning (RL) to address the exposure bias and loss-evaluation mismatch problems for sequence prediction [28, 13]. The self-critical learning approach proposed in [28] is a pioneering work, which well addresses the above two problems.

Our work in this paper is closely related to [35], which uses scene graph to connect images and sentences to incorporate inductive language bias. The key difference is that the framework in [35] is based on the paired setting, while in this work, we learn the scene graph-based network under the unsupervised training setting.

Unpaired Image Captioning. More recently, some researchers started looking into the problem of image captioning in the unpaired setting [14, 10], where there is no correspondence between images and sentences during training. The first work on this task is the pivot-based solution proposed by Gu *et al*. [14]. In their setting, although they do not have any correspondence between images and sentences in the target language, they do require a paired image-caption dataset in the pivot language and another machine translation dataset which consists of sentences in the pivot language and the paired sentences in the target language. They connect the pivot language sentences in different domains by shared word embeddings. The most recent work on unpaired image captioning is proposed by Fang *et al*. [10]. They generate pseudo image-sentence pairs by feeding the visual concepts of images to a concept-to-sentence model and performing the alignment between image features and sentence features in an adversarial manner.

While several attempts have been made for the unpaired image captioning problem, this challenging task is far from mature. Arguably, compared to unpaired sentence-to-sentence [23] and image-to-image [42] translations, unpaired image-to-sentence translation is more challenging because of the significantly different characteristics of the two modalities. In contrast to the existing unpaired image captioning methods [14, 10], our proposed method adopts scene graph as an explicit representation to bridge the gap between image and sentence domains.

3. Method

In this section, we describe our unpaired image captioning framework. We first revisit the paired captioning setting.

3.1. Paired Image Captioning Revisited

In the paired captioning setting, our training goal is to generate a caption S from an image I such that S is similar

to its ground-truth caption. The popular encoder-decoder framework for image captioning can be formulated as:

$$P(S|I) = \underbrace{P(V|I)}_{\text{Encoder}} \underbrace{P(S|V)}_{\text{Decoder}} \quad (1)$$

where the encoder $P(V|I)$ encodes the image I into the image features V with a CNN model [16], and the decoder $P(S|V)$ predicts the image description S from the image features V. The most common training objective is to maximize the probability of the ground-truth caption words given the image: $\sum_t \log p_{\theta_{I\to S}}(S_t|S_{0:t-1}, I)$, where $p_{\theta_{I\to S}}(S_t|S_{0:t-1}, I)$ corresponds to the Softmax output at time step t. During inference, the word S_t is drawn from the dictionary $\mathcal{D}_S$ according to the Softmax distribution.

3.2. Unpaired Image Captioning

In the unpaired image captioning setting, we have a dataset of images $\mathcal{I} = \{I_1, \ldots, I_{N_I}\}$, and a dataset of sentences $\mathcal{S} = \{S_1, \ldots, S_{N_S}\}$, where N_I and N_S are the total numbers of images and sentences, respectively. In this setting, there is no alignment between $\mathcal{I}$ and $\mathcal{S}$. In fact, $\mathcal{I}$ and $\mathcal{S}$ can be completely unrelated coming from two different domains. Our goal is to train an image captioning model in a completely unsupervised way. In our setup, we assume that we have access to an off-the-shelf image scene graph detector and a sentence (or text) scene graph parser.

As shown in Figure 1, our proposed image captioning model consists of one image scene graph generator, one sentence scene graph generator, one scene graph encoder G^S_{Enc}, one attention-based decoder for sentence generation G^S_{Dec}, and a cycle-consistent feature alignment module. Given an image I as input, our method first extracts an image scene graph $\mathcal{G}^I$ using the scene graph generator. It then maps $\mathcal{G}^I$ to the sentence scene graph $\mathcal{G}^S$ from which the RNN-based decoder generates a sentence S. More formally, the image captioner $P(S|I)$ in the unpaired setting can be decomposed into the following submodels,

$$I \to \mathcal{G}^I \quad (2)$$

$$P(S|I) = \underbrace{P(\mathcal{G}^S|\mathcal{G}^I)}_{\text{Unpaired Mapping}} \underbrace{P(S|\mathcal{G}^S)}_{\text{Decoder}} \quad (3)$$

where $\mathcal{G}^I$ and $\mathcal{G}^S$ are the image scene graph and the sentence scene graph, respectively. The most crucial component in Eq. (3) is the unpaired mapping of image and text scene graphs. In our approach, this mapping is done in the feature space. In particular, we encode the image and sentence scene graphs into feature vectors and learn to map the feature vectors across the two modalities. We reformulate Eq. (3) as follows:

$$\begin{aligned} P(S|I) &= P(\mathcal{G}^S|\mathcal{G}^I)P(S|\mathcal{G}^S) \\ &\approx P(\boldsymbol{f}^I|\mathcal{G}^I)P(\boldsymbol{f}^S|\boldsymbol{f}^I)P(S|\boldsymbol{f}^S) \end{aligned} \quad (4)$$

where $P(\boldsymbol{f}^I|\mathcal{G}^I)$ is a graph encoder, $P(S|\boldsymbol{f}^S)$ is an RNN-based sentence decoder, and $P(\boldsymbol{f}^S|\boldsymbol{f}^I)$ is a cross-modal feature mapper in the unpaired setting. In our implementation, we learn the scene graph encoder and the RNN-based decoder on the text modality first, and then we try to map the image scene graph into a common feature space (*i.e.*, the text space) so that the same sentence decoder can be used to decode the sentence from the mapped image features.

The sentence encoding and decoding processes can be formulated as the following two steps:

$$S \to \mathcal{G}^S \quad (5)$$

$$\hat{S} = \arg\max_S P(S|\boldsymbol{f}^S)P(\boldsymbol{f}^S|\mathcal{G}^S) \quad (6)$$

where $\hat{S}$ is the reconstructed sentence. We train the model to enforce $\hat{S}$ to be close to the original sentence S.

In the following, we describe the scene graph generator in Sec. 3.2.1, the scene graph encoder in Sec. 3.2.2, the sentence decoder in Sec. 3.2.3, and our unpaired feature mapping process in Sec. 3.2.4.

3.2.1 Scene Graph Generator

Formally, a scene graph is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ containing a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$. As exemplified in Figure 1, the nodes can be of three types: object node, attribute node, and relationship node. We denote o_i as the i-th object, $r_{i,j}$ as the relation between object o_i and o_j, and a^l_i as the l-th attribute of object o_i.

An image scene graph generator contains an object detector, an attribute classifier, and a relationship classifier. We use Faster-RCNN [27] as the object detector, MOTIFS [39] as the relationship detector, and an additional classifier for attribute identification [35].

To generate the sentence scene graph $\mathcal{G}^S$ for a sentence, we first parse the sentence into a syntactic tree using the parser provided by [2], which uses a syntactic dependency tree built by [21]. Then, we transform the tree into a scene graph with a rule-based method [29].

3.2.2 Scene Graph Encoder

We follow [35] to encode a scene graph. Specifically, we represent each node as a d_e-dimensional feature vector, and use three different spatial graph convolutional encoders to encode the three kinds of nodes by considering their neighborhood information in the scene graph.

Encoding objects. In a scene graph (image or sentence), an object o_i can play either a *subject* or an *object* role in a relation triplet depending on the direction of the edge. Therefore, for encoding objects, we consider what relations they are associated with and what roles they play in that relation. Let $\langle o_i, o_j, r_{i,j}\rangle$ denote the triplet for relation $r_{i,j}$, where o_i

plays a *subject* role and o_j plays an *object* role. The encoding for object o_i, that is $\boldsymbol{x}_{o_i} \in \mathbb{R}^{d_x}$ is computed by

$$\boldsymbol{x}_{o_i} = \frac{1}{N_{r_i}} \sum_{o_j} g_s(\boldsymbol{e}_{o_i}, \boldsymbol{e}_{o_j}, \boldsymbol{e}_{r_{i,j}}) + \frac{1}{N_{r_i}} \sum_{o_k} g_o(\boldsymbol{e}_{o_k}, \boldsymbol{e}_{o_i}, \boldsymbol{e}_{r_{k,i}}) \quad (7)$$

where $\boldsymbol{e}_{o_i} \in \mathbb{R}^{d_e}$ and $\boldsymbol{e}_{r_{i,j}} \in \mathbb{R}^{d_e}$ are the embeddings (randomly initialized) representing the object o_i and the relation $r_{i,j}$, respectively; $g_s(\cdot)$ and $g_o(\cdot)$ are the spatial graph convolution operations for objects as a *subject* and as an *object*, respectively; and N_{r_i} is the total number of relation triplets that o_i is associated with in the scene graph.

Encoding attributes. An object o_i may have multiple attributes in the scene graph. The encoding of an object based on its attributes, *i.e.*, $\boldsymbol{x}_{a_i} \in \mathbb{R}^{d_x}$ is computed by:

$$\boldsymbol{x}_{a_i} = \frac{1}{N_{a_i}} \sum_{l} g_a(\boldsymbol{e}_{o_i}, \boldsymbol{e}_{a_i^l}) \quad (8)$$

where N_{a_i} is the total number of attributes that object o_i has, and $g_a(\cdot)$ is the spatial convolutional operation for attribute based encoding.

Encoding relations. Each relation $r_{i,j}$ is encoded into $\boldsymbol{x}_{r_{i,j}} \in \mathbb{R}^{d_x}$ by considering the objects that the relation connects in the relation triplet,

$$\boldsymbol{x}_{r_{i,j}} = g_r(\boldsymbol{e}_{o_i}, \boldsymbol{e}_{o_j}, \boldsymbol{e}_{r_{i,j}}) \quad (9)$$

where $g_r(\cdot)$ is the associated convolutional operation.

After graph encoding, for each image scene graph or sentence scene graph, we have three sets of embeddings:

$$\begin{aligned} \mathcal{X}_o^k &= [\boldsymbol{x}_{o_1}^k, \cdots, \boldsymbol{x}_{o_{N_o^k}}^k] \\ \mathcal{X}_r^k &= [\boldsymbol{x}_{r_1}^k, \cdots, \boldsymbol{x}_{r_{N_r^k}}^k], \quad k \in \{I, S\} \\ \mathcal{X}_a^k &= [\boldsymbol{x}_{a_1}^k, \cdots, \boldsymbol{x}_{a_{N_a^k}}^k] \end{aligned} \quad (10)$$

where N_o^k, N_r^k, and N_a^k can be different from each other. Figure 2 illustrates the encoding process.

3.2.3 Sentence Decoder

The goal of the sentence decoder is to generate a sentence $\hat{S}$ from the encoded embeddings, $\mathcal{X}_o^k$, $\mathcal{X}_r^k$, and $\mathcal{X}_a^k$. However, these three sets of embeddings are of different lengths and contain different information. Therefore, their importance for the sentence decoding task also vary. In order to compute a relevant context for the decoding task effectively, we use three attention modules, one for each type of embeddings. The attention module g_{Att}^o over $\mathcal{X}_o^k$ is defined as:

$$\boldsymbol{f}_o^k = \sum_{i=1}^{N_o^k} \alpha_i \boldsymbol{x}_{o_i}^k; \quad \alpha_i = \frac{\exp(\boldsymbol{w}_o^T \boldsymbol{x}_{o_i}^k)}{\sum_m \exp(\boldsymbol{w}_o^T \boldsymbol{x}_{o_m}^k)} \quad (11)$$

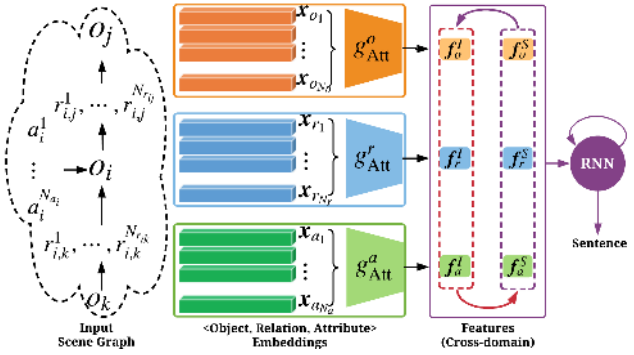


Figure 2: The architectures for scene graph encoding, attention, and sentence decoding; g_{Att}^o, g_{Att}^r, and g_{Att}^a are attention modules for each kind of features, respectively.

where $\boldsymbol{w}_o$ is the associated (learnable) weight vector. The attentions over $\mathcal{X}_r^k$, and $\mathcal{X}_a^k$ are similarly defined to get the respective attention vectors, $\boldsymbol{f}_r^k \in \mathbb{R}^{d_f}$ and $\boldsymbol{f}_a^k \in \mathbb{R}^{d_f}$.

The attention vectors are then combined to get a triplet level embedding, which is then fed into an RNN-based decoder to generate the sentence $\hat{S}$. The following sequence of operations formally describes the process.

$$\boldsymbol{f}_{\text{ora}}^k = g_{\text{ora}}([\boldsymbol{f}_o^k, \boldsymbol{f}_r^k, \boldsymbol{f}_a^k]) \quad (12)$$

$$\mathbf{o}_t, \mathbf{h}_t = \text{RNN}_{\text{Dec}}(\boldsymbol{f}_{\text{ora}}^k, \mathbf{h}_{t-1}, \hat{S}_{t-1}) \quad (13)$$

$$\hat{S}_t \approx \text{softmax}(\mathbf{W}_o \mathbf{o}_t) \quad (14)$$

where $g_{\text{ora}}(\cdot)$ is a neural network that generates a triplet level embedding, and $\mathbf{o}_t$ is the cell output of the decoder at time step t.

3.2.4 Training and Inference

We first train the graph encoder and the sentence decoder in the text modality (Eq. (6)), and then perform a feature level alignments for cross-modal unsupervised mapping.

Training in Text Modality. The graph convolutional encoding of a sentence scene graph $\mathcal{G}^S$ into a feature representation $\boldsymbol{f}_{\text{ora}}^S$, and reconstructing the original sentence S from it are shown at the bottom part of Figure 1, where the encoder and the decoder are denoted as G_{Enc}^S and G_{Dec}^S, respectively. We first train G_{Enc}^S and G_{Dec}^S models by minimizing the cross-entropy (XE) loss:

$$\mathcal{L}_{\text{XE}}(\theta_{\mathcal{G}\to S}) = -\sum_t \log p_{\theta_{\mathcal{G}\to S}}(S_t | S_{0:t-1}) \quad (15)$$

where $\theta_{\mathcal{G}\to S}$ are the parameters of G_{Enc}^S and G_{Dec}^S, $p_{\theta_{\mathcal{G}\to S}}(S_t|S_{0:t-1})$ is the output probability of t-th word in the sentence given by the sentence decoder.

We further employ a reinforcement learning (RL) loss that takes the entire sequence into account. Specifically, we take the CIDEr [32] score as the reward and optimize $\theta_{\mathcal{G}\to S}$

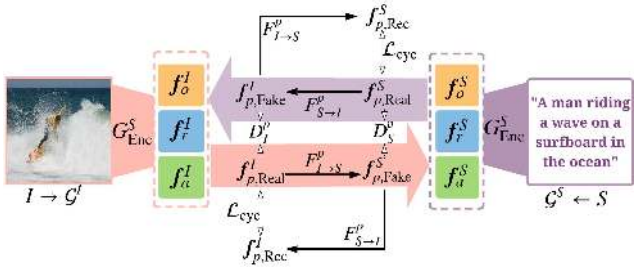


Figure 3: Conceptual illustration of our unpaired feature mapping. For each kind of embedding p, there are two mapping functions $F^p_{I\to S}$ and $F^p_{S\to I}$, and two associated adversarial discriminators D^p_I and D^p_S.

by minimizing the negative expected rewards as follows:

$$\mathcal{L}_{\text{RL}}(\theta_{\mathcal{G}\to S}) = -\mathbb{E}_{\tilde{S}\sim P_{\theta_{\mathcal{G}\to S}}}[r(\tilde{S})] \tag{16}$$

where $r(\tilde{S})$ is the reward calculated by comparing the sampled sentence $\tilde{S}$ with the ground-truth sentence S using the CIDEr metric. In our model, we follow the RL approach proposed in [28, 13].

Unsupervised Mapping of Scene Graph Features. To adapt the learned model from sentence modality to the image modality, we need to translate the scene graph from the image to the sentence modality. We take the discrepancy in the modality of scene graphs directly into account by aligning the representation of the image scene graph with the sentence scene graph. We propose to use CycleGAN [42] to learn the feature alignment across domains.

Figure 3 illustrates our idea. Given two sets of unpaired features $\boldsymbol{f}^I_p$ and $\boldsymbol{f}^S_p$, where $p \in \{o, r, a\}$, we have two mapping functions $F^p_{I\to S}(\cdot)$ and $F^p_{S\to I}(\cdot)$, and two discriminators D^p_S and D^p_I. $F^p_{I\to S}(\cdot)$ maps the image features to the sentence features, and $F^p_{S\to I}(\cdot)$ maps the sentence features to the image features. The discriminators are trained to distinguish the *real* (original modality) features from the *fake* (mapped) features. The mappers are trained to fool the respective discriminators through adversarial training.

For image to text mapping, the adversarial loss is

$$\begin{aligned}\mathcal{L}_{\text{GAN}}(F^p_{I\to S}, D^p_S) &= \mathbb{E}_S[\log D^p_S(\boldsymbol{f}^S_{p,\text{Real}})] \\ &+\mathbb{E}_I[\log(1 - D^p_S(F^p_{I\to S}(\boldsymbol{f}^I_{p,\text{Real}})))]\end{aligned} \tag{17}$$

Similarly, for sentence to image mapping, we have the similar adversarial loss,

$$\begin{aligned}\mathcal{L}_{\text{GAN}}(F^p_{S\to I}, D^p_I) &= \mathbb{E}_I[\log D^p_I(\boldsymbol{f}^I_{p,\text{Real}})] \\ &+\mathbb{E}_S[\log(1 - D^p_I(F^p_{S\to I}(\boldsymbol{f}^S_{p,\text{Real}})))]\end{aligned} \tag{18}$$

Due to the unpaired setting, the mapping from the source to the target modality is highly under-constrained. To make the mapping functions cycle-consistent, CycleGAN introduces a cycle consistency loss to regularize the training,

$$\begin{aligned}\mathcal{L}_{\text{cyc}}(F^p_{S\to I}, F^p_{I\to S}) =&\mathbb{E}_I[\|\boldsymbol{f}^I_{p,\text{Rec}} - \boldsymbol{f}^I_{p,\text{Real}}\|_1] \\ &+\mathbb{E}_S[\|\boldsymbol{f}^S_{p,\text{Rec}} - \boldsymbol{f}^S_{p,\text{Real}}\|_1]\end{aligned} \tag{19}$$

where $\boldsymbol{f}^I_{p,\text{Rec}}$ and $\boldsymbol{f}^S_{p,\text{Rec}}$ are the reconstructed features in the image and text modalities, respectively.

Formally, our overall objective for unpaired feature mapping is to optimize the following loss:

$$\mathcal{L}_{\text{Adv}}(\theta_{I\leftrightarrow S}) = \sum_{p\in\{o,r,a\}} \mathcal{L}_{\text{Adv}}(\theta^p_{I\leftrightarrow S}) \tag{20}$$

$$\begin{aligned}\mathcal{L}_{\text{Adv}}(\theta^p_{I\leftrightarrow S}) &= \mathcal{L}_{\text{GAN}}(F^p_{S\to I}, D^p_I) + \mathcal{L}_{\text{GAN}}(F^p_{I\to S}, D^p_S) \\ &+ \lambda\mathcal{L}_{\text{cyc}}(F^p_{S\to I}, F^p_{I\to S})\end{aligned} \tag{21}$$

where $\theta^p_{I\leftrightarrow S}$ are the parameters of the two mapping functions and the discriminators for each kind of embedding p, and λ is a hyperparameter to control the regularization.

Cross-modal Inference. During inference, given an image I, we generate its corresponding scene graph $\mathcal{G}^I$ using a pre-trained image scene graph generator, use the scene graph encoder to get the image features $\boldsymbol{f}^I_p$, which are then mapped through the image-to-text mapper $F^p_{I\to S}$. The mapped features are then used for sentence generation using the sentence decoder. The cross-modal inference process can be formally expressed as:

$$\hat{S} = \arg\max_S P(S|\boldsymbol{f}^I_{\text{ora}})P(\boldsymbol{f}^I_r, \boldsymbol{f}^I_o, \boldsymbol{f}^I_a|\mathcal{G}^I) \tag{22}$$

$$\boldsymbol{f}^I_{\text{ora}} = g_{\text{ora}}([F^o_{I\to S}(\boldsymbol{f}^I_{\text{o}}), F^r_{I\to S}(\boldsymbol{f}^I_{\text{r}}), F^a_{I\to S}(\boldsymbol{f}^I_{\text{a}})]) \tag{23}$$

where $g_{\text{ora}}(\cdot)$ is the same module as Eq. (12).

4. Experiments

In this section, we evaluate the effectiveness of our proposed method. We first introduce the datasets and the experimental settings. Then, we present the performance comparisons as well as ablation studies to understand the impact of different components of our framework.

4.1. Datasets and Setting

Table 1 shows the statistics of the training datasets used in our experiments. We use Visual Genome (VG) dataset [22] to train our image scene graph generator. We filter the object, attribute, and relation annotations by keeping those that appear more than 2,000 times in the training set. The resulting dataset contains 305 objects, 103 attributes, and 64 relations (a total of 472 items).

We collect the image descriptions from the training split of MSCOCO [24] and use them as our sentence corpus to train the scene graph encoder and the sentence decoder. In pre-processing, we tokenize the sentences and convert all the tokens to lowercase. The tokens that appear less than

five times are treated as ⟨UNK⟩ tokens. The maximum caption length is fixed to 16, and all the captions longer than 16 are truncated. This results in a base vocabulary of 9,487 words. For sentence scene graph generation, we generate the scene graph using the language parser in [2, 35]. We perform a filtering process by removing objects, relations, and attributes which appear less than 10 times in all the parsed scene graphs. After this filtering, we obtain 5,364 objects, 1,308 relations, and 3,430 attributes. This gives an extended vocabulary where the previous 9,487 words are consistent with the base vocabulary. The embeddings for the vocabulary items are randomly initialized.

Table 1: Statistics of the training datasets.

Scene Graph	Vocabulary Size		
	#Object	#Attribute	#Relation
Image (VG)	305	103	64
Sentence (MSCOCO)	5,364	3,430	1,308

For learning the mapping between the modalities, the unpaired training data is intentionally collected by shuffling the images and the sentences from MSCOCO randomly. We validate the effectiveness our method on the same test splits as used in [14, 10] for a fair comparison. The widely used CIDEr-D [32], BLEU [26], METEOR [5], and SPICE [2] are used to measure the quality of the generated captions.

4.2. Implementation Details

We follow [35] to train our image scene graph generator on VG. We first train a Faster-RCNN and use it to identify the objects in each image. We select at least 10 and at most 100 objects for an image. The object features extracted by RoI pooling are used as input to the object detector, the relation classifier, and the attribute classifier. We adopt the LSTM-based relation classifier from [39]. Our attribute classifier is a single hidden layer network with ReLU activation (*i.e.*, fc-ReLU-fc-Softmax), and we keep only the three most probable attributes for each object. For scene graph encoding, we set $d_e = d_x = d_f = 1000$. We implement g_s, g_o, g_r, g_a, and g_{ora} (Eq. (7) - (12)) as fully-connected layers with ReLU activations. The two mapping functions in Eq. (17) and Eq. (18) are implemented as fully-connected layers with leaky ReLU activations.

The sentence decoder has two LSTM layers. The input to the first LSTM is the word embeddings and its previous hidden state. The input to the second LSTM is the concatenation of three terms: the triplet embedding $\boldsymbol{f}_{\text{ora}}$, the output from the first LSTM, and its previous hidden state. We set the number of hidden units in each LSTM to 1,000.

During training, we first train the network with the cross-entropy loss (Eq. (15)) for 20 epochs and then fine-tune it with RL loss in Eq. (16). The learning rate is initialized to 4×10^{-4} for all parameters and decayed by 0.8 after every 5 epoch. We use Adam [20] for optimization with a batch size of 50. During the (unpaired) alignment learning, we freeze the parameters of the scene graph encoder and the sentence decoder, and only learn the mapping functions and the discriminators. For all the experiments, we empirically set λ to 10 in Eq. (21). During inference, we use beam search with a beam size of 5.

For quantifying the efficacy of the proposed framework, we use several baselines for performance comparison.

Graph-Enc-Dec (Avg). This baseline learns the graph encoder G^S_{Enc} and the sentence decoder G^S_{Dec} only on sentence corpus. It takes the average operation (as opposed to attention) over the three sets of features: $\mathcal{X}^k_o$, $\mathcal{X}^k_r$, and $\mathcal{X}^k_a$. During testing, we directly feed the image scene graph $\mathcal{G}^I$ to this model and get the image description.

Graph-Enc-Dec (Att*). This model shares the same setting with Graph-Enc-Dec (Avg) but replaces the average operation with a shared attention mechanism for all three sets (i.e., same attention for object, attribution, and relation).

Graph-Enc-Dec (Att). This model modifies the Graph-Enc-Dec (Att*) with an independent attention mechanism for each set of features.

Graph-Align. This is our final model. It is initialized with the trained parameters from Graph-Enc-Dec (Att) that uses separate attentions, and then it also learns the feature mapping functions using adversarial training.

4.3. Quantitative Results

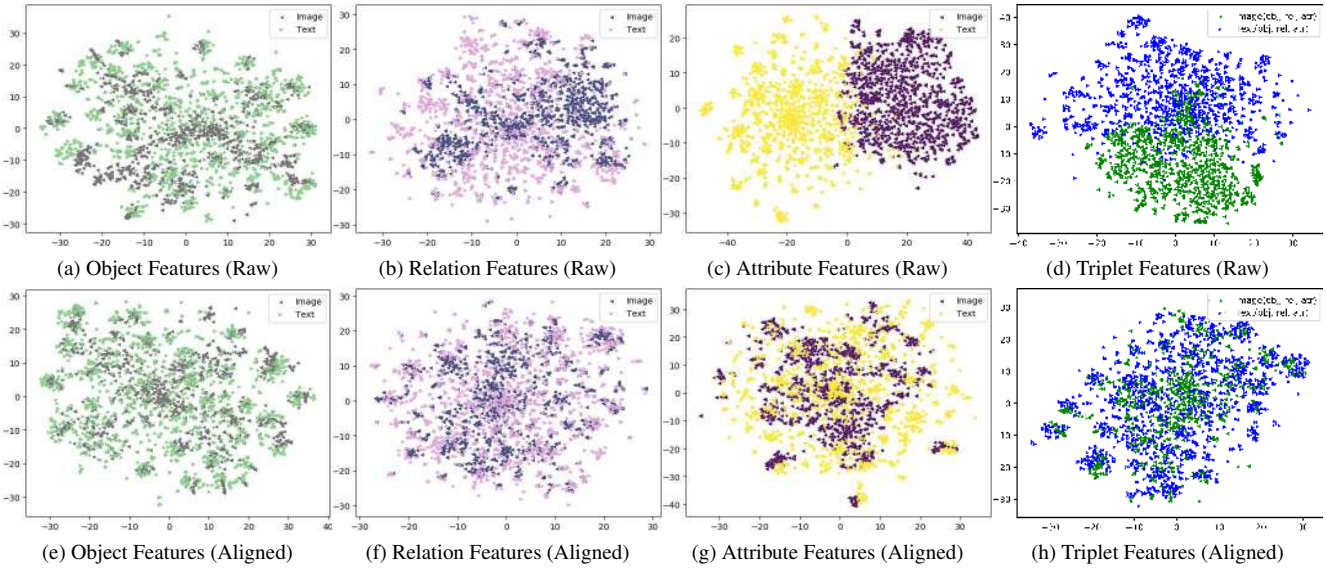
Investigation on Sentence Decoding. In this experiment, we first train the network with Eq. (15), and then fine-tune it with Eq. (16) on the sentence corpus. Table 2 compares the results of three baseline models on the sentence corpus. It can be seen that the attention-based model performs better than the average-based model in all metrics, which demonstrates that weighting over features can better model the global dependency of features. Note that separate attention model for each set of features can significantly improve the performance. The inconsistent alignment of three kinds of features in Figure 4 also supports that we should treat these sets of features separately.

Table 2: Results for different sentence scene graph decoders on MSCOCO test split, where B@n refers to BLEU-n, M refers to METEOR, and C refers to CIDEr. All values are reported in percentage (bold numbers are the best results).

Methods	B@1	B@2	B@3	B@4	M	C
Graph-Enc-Dec(Avg)	84.3	71.8	58.8	47.1	31.0	129.4
Graph-Enc-Dec(Att*)	91.8	80.3	67.5	55.5	34.3	151.4
Graph-Enc-Dec(Att)	94.1	84.6	72.9	61.5	36.3	168.8

Table 3: Results for different baselines without GAN training on the test split of the MSCOCO.

Methods	B@1	B@2	B@3	B@4	M	C
Graph-Enc-Dec(Avg)	52.1	34.1	23.8	17.6	14.9	41.4
Graph-Enc-Dec(Att*)	54.3	37.0	26.8	20.3	15.9	47.2
Graph-Enc-Dec(Att)	56.0	33.6	20.1	11.9	17.0	48.5



(a) Object Features (Raw) (b) Relation Features (Raw) (c) Attribute Features (Raw) (d) Triplet Features (Raw)

(e) Object Features (Aligned) (f) Relation Features (Aligned) (g) Attribute Features (Aligned) (h) Triplet Features (Aligned)

Figure 4: Visualization of features in 2D space by t-SNE [25]. We plot the scatter diagrams for 1,500 samples.

Investigation on Unpaired Setting without GAN. Table 3 shows the comparisons among different baselines when no explicit cross-modal mapping of the features is done. By feeding the image scene graph directly to the trained scene graph encoder and the sentence decoder, we can achieve promising performance on the test set. Graph-Enc-Dec (Att) still achieves the best performance in all metrics. This is reasonable since both scene graphs and captions are high-level understandings of the image, and by capturing rich semantic information about objects and their relationships, scene graphs provide an effective way to connect an image to its natural language description. This finding also validates the feasibility of our approach to unpaired image captioning through the use of scene graphs. However, compared to the paired setup (see Table 6), these results are still inferior, meaning that only scene graph is not enough to achieve comparable performance.

Investigation on Unpaired Setting with GAN. To align the features from the image modality to the text modality, we use CycleGAN with our Graph-Enc-Dec(Att) model. Table 4 shows the comparisons of three kinds of GAN loss: binary cross-entropy (BCE) loss with logits (the vanilla GAN loss [11]), mean squared error (MSE) loss, and gradient penalty (GP) [18]. We also compare the results for using different output dimensions in the discriminator.[1]

We can see that most of the CycleGAN variants improve the performance substantially compared to the results in Table 3. The GP with 64-dimension discriminator output achieves the best performance. Note that, when we set the output dimension to 1, the performance drops. This indicates that a strong discriminator is crucial for unpaired feature alignments. From the bottom row of Figure 4, we can see that with the help of the mapping module, the three kinds of embeddings are aligned very well, especially the attribute embedding (Figure 4g). It is also worth noting that the triplet features in Figure 4h are better aligned compared to the raw triplet features in Figure 4d.

To further demonstrate the effectiveness of the proposed three feature mapping functions, we conduct additional experiments in Table 5. It can be seen that treating the three set of embeddings ($\mathcal{X}_o^k$, $\mathcal{X}_r^k$, and $\mathcal{X}_a^k$) without distinction performs worse than Graph-Align.

[1]For example, for a dimension of 64, the output is a 64-dimensional vector, which is compared against an all-one vector of length 64 for a 'Real' input, and with an all-zero vector of length 64 for a 'Fake' input.

Table 4: Ablation studies of different GAN losses for Graph-Align model.

GAN Loss	Discriminator	B@1	B@2	B@3	B@4	C
BCE	$d_f \to d_f$	64.9	44.2	28.6	18.1	63.0
	$d_f \to 64$	66.0	46.0	30.3	19.7	65.5
	$d_f \to 1$	65.5	45.4	29.6	18.8	65.2
MSE	$d_f \to d_f$	65.3	44.8	28.9	18.3	62.9
	$d_f \to 64$	66.0	45.9	29.7	18.8	63.8
	$d_f \to 1$	58.4	36.3	21.7	12.6	46.7
GP	$d_f \to d_f$	66.1	46.1	30.3	19.5	65.5
	$d_f \to 64$	**67.1**	**47.8**	**32.3**	**21.5**	**69.5**
	$d_f \to 1$	64.5	44.2	28.5	17.9	61.1

Table 5: The performances of using different feature mappings on MSCOCO test split. Shared GAN learns a shared feature mapping for three sets of features with CycleGAN. Single GAN concatenates the three kinds of embeddings together and learns a mapping with CycleGAN.

Methods	B@1	B@2	B@3	B@4	M	C
Shared GAN	60.7	41.3	26.9	17.6	20.0	60.1
Single GAN	61.8	42.1	27.3	17.7	20.1	61.2
Graph-Align	**67.1**	**47.8**	**32.3**	**21.5**	**20.9**	**69.5**

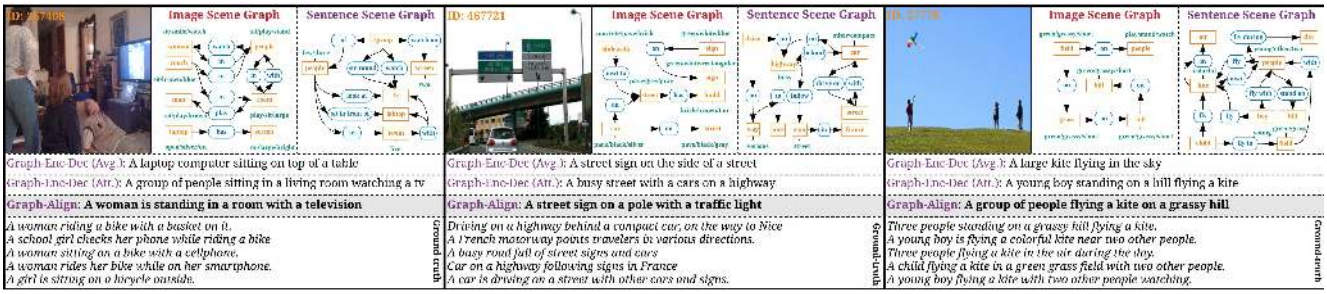


Figure 5: Qualitative examples of different methods. In each example, the left image is the original input image; the middle is the image scene graph; the right image is the ground-truth sentence scene graph for compassion.

Table 6: Performance comparisons on the test split of the MSCOCO dataset.

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE	CIDEr	SPICE
Paired Setting								
SCST [28]	–	–	–	33.3	26.3	55.3	111.4	–
Stack-Cap [13]	78.6	62.5	47.9	36.1	27.4	56.9	120.4	20.9
SGAE (base) [36]	79.9	–	–	36.8	27.7	57.0	120.6	20.9
Unpaired Setting								
Language Pivoting [14]	46.2	24.0	11.2	5.4	13.2	–	17.7	
Adversarial+Reconstruction [10]	58.9	40.3	27.0	18.6	17.9	43.1	54.9	11.1
Graph-Align	**67.1**	**47.8**	**32.3**	**21.5**	**20.9**	**47.2**	**69.5**	**15.0**

Finally, Table 6 compares the results of the Graph-Align model with those of the existing unpaired image captioning methods [14, 10] on the MSCOCO test split. We can notice that our proposed Graph-Align achieves the best performance in all metrics. This demonstrates the effectiveness of our scene graph-based unpaired image captioning model.

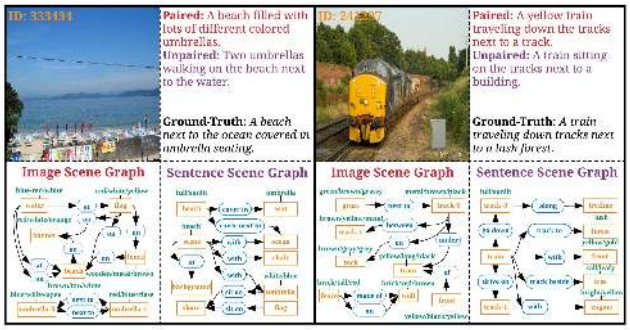


Figure 6: Examples of unpaired image captioning failure cases. Although the accuracy of image scene graph highly influences the performance of captioning results, our Graph-Align can still generate relevant image captions.

4.4. Qualitative Results

Figure 5 visualizes some examples of our models. We show the generated image descriptions using different models along with the ground-truth captions (bottom part). In the generated image and sentence scene graphs, we mark object, relation, attribute nodes in orange, blue, and green, respectively. From these exemplary results, we observe that our method can generate reasonable image descriptions by aligning the unpaired visual-textual modalities with the help of scene graphs. Also, we observe that the number of attributes (words in green) in the sentence scene graph is less than that in the image scene graph. This observation potentially explains why there is a huge feature embedding gap between image and text in Figure 4c.

Figure 6 presents some failure cases of our Graph-Align model. We can see that the image scene graphs mainly focus on local regions/objects, while sentence scene graphs convey more information about the images. Such information misalignment leads to generating different captions.

5. Conclusions

In this paper, we have proposed a novel framework to train an image captioning model in an unsupervised manner without using any paired image-sentence data. Our method uses scene graph as an intermediate representation of the image and the sentence, and maps the scene graphs in their feature space through cycle-consistent adversarial training. We used graph convolution and attention methods to encode the objects, their attributes, their relationships in a scene graph. Our experimental results based on quantitative and qualitative evaluations show the effectiveness of our method in generating meaningful captions, which also outperforms existing methods by a good margin. In future, we would like to evaluate our method on other datasets and explore other mapping methods such as optimal transport.

Acknowledgments

This work was supported in part by NTU-IGS, NTU-Alibaba Lab, NTU DSAIR Center, and NTU ROSE Lab.

References

[1] English-speaking world, 2019. [Online; accessed 22-March-2019]. 1

[2] P. Anderson, B. Fernando, M. Johnson, and S. Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, 2016. 3, 6

[3] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018. 2

[4] M. Artetxe, G. Labaka, E. Agirre, and K. Cho. Unsupervised neural machine translation. In *ICLR*, 2018. 1

[5] S. Banerjee and A. Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL Workshop*, 2005. 6

[6] H. Ding, X. Jiang, A. Q. Liu, N. M. Thalmann, and G. Wang. Boundary-aware feature propagation for scene segmentation. In *ICCV*, 2019. 2

[7] H. Ding, X. Jiang, B. Shuai, A. Q. Liu, and G. Wang. Context contrasted feature and gated multi-scale aggregation for scene segmentation. In *CVPR*, 2018. 2

[8] H. Ding, X. Jiang, B. Shuai, A. Q. Liu, and G. Wang. Semantic correlation promoted shape-variant context for segmentation. In *CVPR*, 2019. 2

[9] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, et al. From captions to visual concepts and back. In *CVPR*, 2015. 1

[10] Y. Feng, L. Ma, W. Liu, and J. Luo. Unsupervised image captioning. In *CVPR*, 2019. 1, 2, 6, 8

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014. 7

[12] J. Gu, J. Cai, S. Joty, L. Niu, and G. Wang. Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models. In *CVPR*, 2018. 1, 2

[13] J. Gu, J. Cai, G. Wang, and T. Chen. Stack-captioning: Coarse-to-fine learning for image captioning. In *AAAI*, 2017. 1, 2, 5, 8

[14] J. Gu, S. Joty, J. Cai, and G. Wang. Unpaired image captioning by language pivoting. In *ECCV*, 2018. 1, 2, 6, 8

[15] J. Gu, G. Wang, J. Cai, and T. Chen. An empirical study of language cnn for image captioning. In *ICCV*, 2017. 2

[16] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al. Recent advances in convolutional neural networks. *Pattern Recognition*, pages 354–377, 2017. 1, 3

[17] J. Gu, H. Zhao, Z. Lin, S. Li, J. Cai, and M. Ling. Scene graph generation with external knowledge and image reconstruction. In *CVPR*, 2019. 2

[18] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *NIPS*, 2017. 7

[19] J. Hitschler, S. Schamoni, and S. Riezler. Multimodal pivots for image caption translation. In *ACL*, 2016. 2

[20] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[21] D. Klein and C. D. Manning. Accurate unlexicalized parsing. In *ACL*, 2003. 3

[22] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017. 5

[23] G. Lample, A. Conneau, L. Denoyer, and M. Ranzato. Unsupervised machine translation using monolingual corpora only. In *ICLR*, 2018. 1, 2

[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 5

[25] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *JMLR*, 2008. 7

[26] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 6

[27] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 3

[28] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. In *CVPR*, 2017. 2, 5, 8

[29] S. Schuster, R. Krishna, A. Chang, L. Fei-Fei, and C. D. Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *ACL*, 2015. 3

[30] K. Shuster, S. Humeau, H. Hu, A. Bordes, and J. Weston. Engaging image captioning via personality. In *CVPR*, 2019. 2

[31] D. Teney, L. Liu, and A. van den Hengel. Graph-structured representations for visual question answering. In *CVPR*, 2017. 2

[32] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015. 4, 6

[33] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *PAMI*, 2017. 1, 2

[34] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 2

[35] X. Yang, K. Tang, H. Z. Zhang, and J. Cai. Auto-encoding scene graphs for image captioning. In *CVPR*, 2019. 2, 3, 6

[36] X. Yang, H. Zhang, and J. Cai. Shuffle-then-assemble: Learning object-agnostic visual relationship features. In *ECCV*, 2018. 2, 8

[37] X. Yang, H. Zhang, and J. Cai. Learning to collocate neural modules for image captioning. 2019. 1

[38] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo. Image captioning with semantic attention. In *CVPR*, 2016. 2

[39] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, 2018. 3, 6

[40] H. Zhao, Z. Ding, and Y. Fu. Multi-view clustering via deep matrix factorization. In *AAAI*, 2017. 2

[41] H. Zhao, Q. Fan, D. Gutfreund, and Y. Fu. Semantically guided visual question answering. In *WACV*, 2018. 2

[42] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 2, 5