Cross-modal Scene Graph Matching for Relationship-aware Image-Text Retrieval

Sijin Wang[1,2], Ruiping Wang[1,2], Ziwei Yao[1,2], Shiguang Shan[1,2], Xilin Chen[1,2]
[1]Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing, 100190, China
[2]University of Chinese Academy of Sciences, Beijing, 100049, China
{sijin.wang, ziwei.yao}@vipl.ict.ac.cn, {wangruiping, sgshan, xlchen}@ict.ac.cn

Abstract

*Image-text retrieval of natural scenes has been a popular research topic. Since image and text are heterogeneous cross-modal data, one of the key challenges is how to learn comprehensive yet unified representations to express the multi-modal data. A natural scene image mainly involves two kinds of visual concepts, objects and their relationships, which are equally essential to image-text retrieval. Therefore, a good representation should account for both of them. In the light of recent success of scene graph in many CV and NLP tasks for describing complex natural scenes, we propose to represent image and text with two kinds of scene graphs: visual scene graph (**VSG**) and textual scene graph (**TSG**), each of which is exploited to jointly characterize objects and relationships in the corresponding modality. The image-text retrieval task is then naturally formulated as cross-modal scene graph matching. Specifically, we design two particular scene graph encoders in our model for VSG and TSG, which can refine the representation of each node on the graph by aggregating neighborhood information. As a result, both object-level and relationship-level cross-modal features can be obtained, which favorably enables us to evaluate the similarity of image and text in the two levels in a more plausible way. We achieve state-of-the-art results on Flickr30k and MS COCO, which verifies the advantages of our graph matching based approach for image-text retrieval.*

1. Introduction

Visual media and natural language are the two most prevalent information coming in different modalities in our daily life. To achieve artificial intelligence on computers, it is essential to enable computers to understand, match, and transform such cross-modal data. Image-text cross-modal retrieval is thus one of the challenging research topics, where given a query of one modality (an image or a text sentence), it aims to retrieve the most similar samples from the database in another modality. The key challenge here is how to match the cross-modal data by understanding their contents and measuring their semantic similarity, especially when there are multiple objects in the cross-modal data.

To address this task, many approaches have been proposed. As shown in the top of Fig.1, early approaches [14, 3, 27, 28, 38] use global representations to express the whole image and sentence, which ignore the local details. Such approaches work well on simple cross-modal retrieval scenario that contains only a single object, but are not satisfactory for more realistic cases that involve complex natural scenes. Recent studies [12, 11, 7, 8, 17] pay attention to local detailed matching by detecting objects in both images and text, and have gained certain improvements over previous works, which is described in the middle of Fig.1.

However, a natural scene contains not only several objects but also their relationships [10], which are equally important to image-text retrieval. For example, three images in the left of Fig.1 contain similar objects. The *"dog"* in $img1$ can distinguish this image from the other two, while $img2$ and $img3$ contain the same objects, including *"woman"*, *"horse"*, *"beach"* and *"dress"*. To discriminate such two images, the relationships play an essential role. Clearly, the *"woman"* in $img2$ is *"standing next to"* the horse while the *"woman"* in $img3$ is *"riding on"* the horse. Similarly, there are also semantic relationships between textual objects in a sentence after syntactic analysis, such as *"woman-wears-dress"*, *"woman-rides-on-horse"* in the text query in Fig.1.

With more recent research topics focusing on the objects and relationships in the image scene, scene graphs [10] are proposed to model the objects and relationships formally and have quickly become a powerful tool used in high-level semantic understanding tasks [18, 35, 9, 29, 34]. A scene graph consists of many nodes and edges, in which each node represents an object, and each edge indicates the relationship between the two nodes it connects. To represent

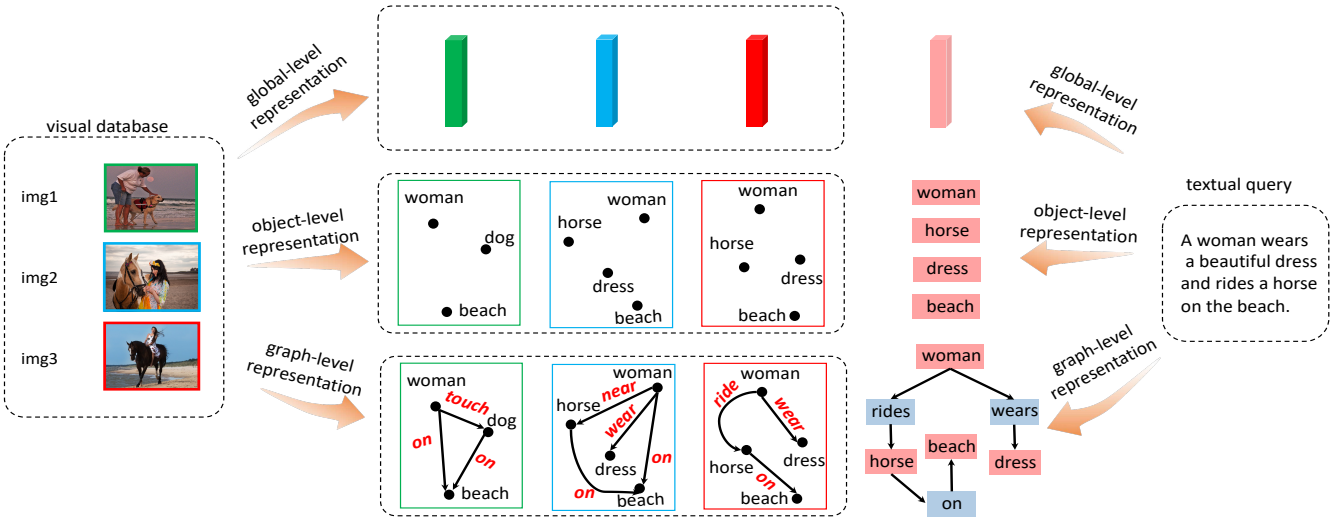

Figure 1. Three different image-text retrieval frameworks. The framework on the top uses global representations to present images and text for matching. The middle one extracts objects in the image and text for detailed matching. The bottom one (ours) captures both objects and their relationships from the image and text with two graphs for two levels matching.

the image and text comprehensively in the image-text retrieval task, we organize the objects and the relationships into scene graphs for both modalities, as illustrated in the bottom of Fig.1. We introduce a visual scene graph (VSG) and a textual scene graph (TSG) to represent images and text, respectively, converting the conventional image-text retrieval problem to the matching of two scene graphs.

To be specific, we extract objects and relationships from the image and text to form the VSG and TSG, and design a so-called Scene Graph Matching (**SGM**) model, where two tailored graph encoders encode the VSG and TSG into the visual feature graph and the textual feature graph. The VSG encoder is a Multi-modal Graph Convolutional Network (MGCN), which enhances the representations of each node on the VSG by aggregating useful information from other nodes and updates the object and relationship features in different manners. The TSG encoder contains two different bi-GRUs aiming to encode the object and relationship features, respectively. After that, both object-level and relationship-level features are learned in each graph, and the two feature graphs corresponding to two modalities can be finally matched at two levels in a more plausible way.

To evaluate the effectiveness of our approach, we conduct image-text retrieval experiments on two challenging datasets, Flickr30k [36] and MS COCO [19]. The results show that the performance of our approach significantly outperforms state-of-the-art methods and validates the importance of relationships for image-text retrieval.

2. Related Works

Image-Text Retrieval. Image-text retrieval task has become a popular research topic in recent years. Several excellent works [14, 3, 24, 27, 12, 11, 7, 8, 17, 15, 38, 5] are introduced to address this task, which can be divided into two groups: i) global representation based methods and ii) local representation based methods.

Global representation based methods [3, 27, 28, 38, 4, 14] usually consist of an image encoder (*e.g.* CNN) and a sentence encoder (*e.g.* RNN) to extract a global feature of the image and sentence, respectively. Then, a metric is devised to measure the similarity of a couple of features in different modalities. Frome *et al.* [4] proposed a deep visual semantic embedding model that uses CNN to extract the visual representations from the full image and Skip-Gram [20] to obtain the representation of the semantic labels. Similarly, Kiros *et al.* [14] use LSTM to encode the full sentence and the triplet loss to make the matched image-sentence pair closer than the unmatched pairs in the embedding space. Wehrmann *et al.* [31] designed an efficient character-level inception module which encodes textual features by convolving raw characters in the sentence. Faghri *et al.* [3] produce significant gains in retrieval performance by introducing hard negatives mining into triplet loss.

To be more detailed, local representation based methods [12, 11, 7, 8, 17] that focus on the local alignment between images and sentences, have been developed recently. Karpathy *et al.* [12] extract objects from images, and match these visual objects with words in the sentences. To improve such an approach, Lee *et al.* [17] attend more important fragments (words or regions) with an attention network. Huang *et al.* [8] propose that semantic concepts, as well as the order of semantic concepts, are essential for image-text matching. To solve the issue of embedding polysemous instances, Song and Soleymani [25] extract K embeddings of

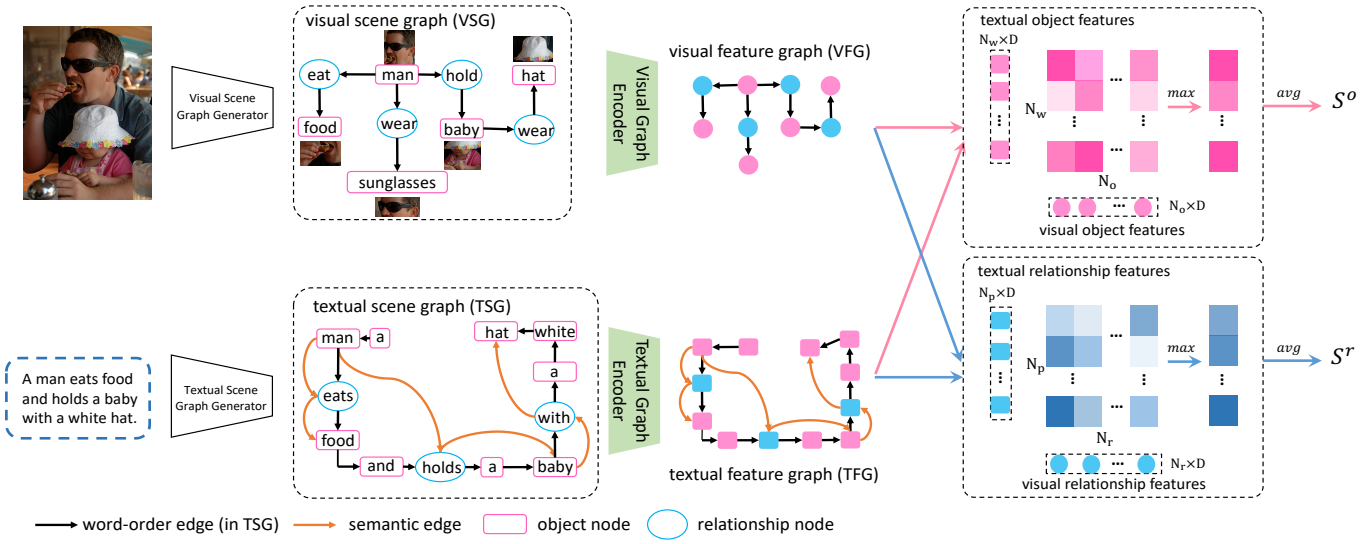

Figure 2. The architecture of our method. The image and sentence are parsed into a VSG and TSG by two graph generators. Then two encoders encode them into feature graphs, which are matched at object and relationship levels at last. Note that VSG and TSG have two types of nodes and TSG contains two kinds of edges that are explained in the legend.

each image rather than injective embedding.

However, some of the above methods lose sight of the relationships between objects in multi-modal data, which is also the key point for image-text retrieval. Though some of them [11, 7, 8, 17] use RNNs to embed words with context, it still does not explicitly reveal the semantic relationships between textual objects. In our approach, both visual and textual objects with their relationships are explicitly captured by scene graphs. Thus, the cross-modal data can match in two levels, which is more plausible.

Scene Graph. Scene graph was first proposed by [10] for image retrieval, which describes objects, their attributes, and relationships in images with a graph. With recent breakthroughs in scene graph generation [37, 18, 33, 30, 30], many high-level visual semantic tasks are developed, such as VQA [26], image captioning [34, 35, 18], and grounding referring expressions [29]. Most of these methods benefit from the use of scene graphs to present images. On the other hand, several methods [1, 30, 22] are proposed to parse the sentence into a scene graph, which is applied to some cross-modal tasks [34]. In recent years, there are attempts to use graph structures to represent both visual and textual data, such as [26] that employs graphs to represent image and text questions for VQA. Distinctive from our method, their graphs, which contain no semantic relationships, are not the so-called scene graph.

3. Method

Given a query in one modality (a sentence query or an image query), the goal of the image-text cross-modal retrieval task is to find the most similar sample from the database in another modality. Therefore, our Scene Graph Matching (SGM) model aims to evaluate the similarity of the image-text pairs by dissecting the input image and text sentence into scene graphs. The framework of SGM is illustrated in Fig.2, which consists of two branches of networks. In the visual branch, the input image is represented into a visual scene graph (VSG) and then encoded into the visual feature graph (VFG). Simultaneously, the sentence is parsed into a textual scene graph (TSG) and then encoded into the textual feature graph (TFG) in the textual branch. Finally, the model collects object features and relationship features from the VFG and TFG and calculates the similarity score at the object-level and relationship-level, respectively. The architectures of the submodules of SGM will be detailed in the following subsections.

3.1. Visual Feature Embedding

3.1.1 Visual Scene Graph Generation

Given a raw image, the visual scene graph is generated by an off-the-shelf scene graph generation method, such as MSDN [18] and Neural Motifs [37]. We represent a visual scene graph as $G = \{V, E\}$, where V is the node-set, and E is the edge-set. There are two types of nodes in our visual scene graph, as shown in Fig.2. The pink rectangles denote object nodes, each of which corresponds to a region of the image. The ellipses in light blue are relationship nodes, each of which connects two object nodes by directed edges. Additionally, each node has a category label, such as *"man", "hold"*.

Concretely, suppose there are N_o object nodes and N_r relationship nodes in a VSG. The object nodes set can be represented as $O = \{o_i | i = 1, 2, \ldots, N_o\}$. The set of rela-

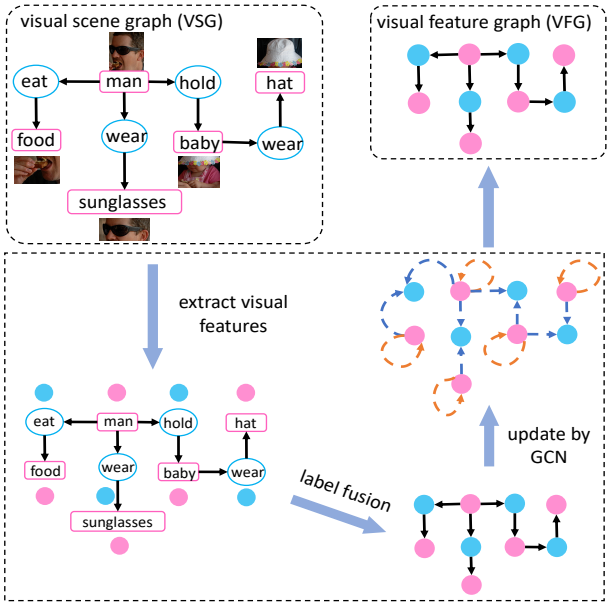

Figure 3. The framework of the VSG encoder. The corresponding image region of each node is embedded into a feature vector by the visual feature extractor. Then the visual feature and the word label of each node are fused by the multi-modal fusion layer. Finally, the graph is encoded by a GCN, and yield the visual feature graph as output.

tionship nodes is $R = \{r_{ij}\} \subseteq O \times O$, where $|R| = N_r$, and r_{ij} is the relationship of o_i and o_j. The label of o_i and r_{ij} can be represented by one-hot vectors, $\mathbf{l}_{o_i}$ and $\mathbf{l}_{r_{ij}}$.

3.1.2 Visual Scene Graph Encoder

After the generation of visual scene graph, we design a *Multi-modal Graph Convolutional Network* (MGCN) to learn good representations on VSG, which includes a pre-trained visual feature extractor, a label embedding layer, a multi-modal fusion layer, and a graph convolutional network, shown in Fig.3.

Visual Feature Extractor. The pre-trained visual feature extractor is used for encoding image regions into feature vectors, which can be pre-trained CNN networks or object detectors (*e.g.* Faster-RCNN [21]). Each node in the VSG will be encoded into a d_1-dimension visual feature vector by the extractor. For object node o_i, its visual feature vector $\mathbf{v}_{o_i}$ is extracted from its corresponding image region. For relationship node r_{ij}, its visual feature vector $\mathbf{v}_{r_{ij}}$ is extracted from the union image region of o_i and o_j.

Label Embedding Layer. Each node has a word label predicted by the visual scene graph generator, which can provide the auxiliary semantic information. The label embedding layer is built to embed the word label of each node into a feature vector. Given the one-hot vectors $\mathbf{l}_{o_i}$ and $\mathbf{l}_{r_{ij}}$, the embedded label features $\mathbf{e}_{o_i}$ and $\mathbf{e}_{r_{ij}}$ are computed as $\mathbf{e}_{o_i} = \mathbf{W}_o\mathbf{l}_{o_i}$ and $\mathbf{e}_{r_{ij}} = \mathbf{W}_r\mathbf{l}_{r_{ij}}$, where $\mathbf{W}_o \in R^{d_2 \times C_o}$ and $\mathbf{W}_r \in R^{d_2 \times C_r}$ are trainable parameters and initialized by word2vec (we use d_2=300). C_o is the category number of objects and C_r is the category number of relationships.

Multi-modal Fusion Layer. After obtaining the visual feature and label feature of each node, it is necessary to fuse them into a unified representation. Thus, a multi-modal fused feature graph is generated. Specifically, the visual feature and label feature are concatenated, then fused as

$$\mathbf{u}_{o_i} = \tanh(\mathbf{W}_u[\mathbf{v}_{o_i}, \mathbf{e}_{o_i}]), \quad (1)$$

$$\mathbf{u}_{r_{ij}} = \tanh(\mathbf{W}_u[\mathbf{v}_{r_{ij}}, \mathbf{e}_{r_{ij}}]), \quad (2)$$

where $\mathbf{W}_u \in R^{d_1 \times (d_1+d_2)}$ is the trainable parameter of the fusion layer.

Graph Convolutional Network. GCNs [32] are convolutional neural networks that can operate on graphs of any structure, which is more flexible than CNNs that can only work on grid structured data. To encode the multi-modal fused feature graph, we adopt an m-layer GCN and propose a novel update mechanism to update two kinds of nodes in different manners. The object nodes will generate object-level features, which can be seen as the first-order features of the image. It may ruin the representation of the object node by the information from another object node or relationship node so that each object node is updated without other information from the neighborhoods. On the contrary, the relationship-level features are the second-order features of the image, so the representations of relationship nodes can be enhanced by its adjacent object nodes. Therefore, relationship nodes update by aggregating information from their neighborhoods and object nodes update from themselves, as shown by the blue and yellow dashed arrows in Fig.3. Concretely, given the multi-modal fused feature graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ (distinguished from the raw visual scene graph $G = \{V, E\}$), the k-th layer of GCN is computed as

$$\mathbf{h}^k_{o_i} = g_o(\mathbf{h}^{k-1}_{o_i}), \qquad \mathbf{h}^k_{r_{ij}} = g_r(\mathbf{h}^{k-1}_{o_i}, \mathbf{h}^{k-1}_{r_{ij}}, \mathbf{h}^{k-1}_{o_j}), \quad (3)$$

where g_r and g_o are fully-connected layers, followed by a tanh function. The initial hidden states are the fused features as $\mathbf{h}^0_{o_i} = \mathbf{u}_{o_i}$ and $\mathbf{h}^0_{r_{ij}} = \mathbf{u}_{r_{ij}}$.

Finally, the output of an m-layer GCN is an encoded visual feature graph with two kinds of vertices: $\mathbf{h}_{o_i}$, $\mathbf{h}_{r_{ij}}$.

3.2. Textual Feature Embedding

3.2.1 Textual Scene Graph Generation

Similar to images, a natural language sentence also describes many objects and their relationships. Therefore, the graph structure is also appropriate for representing a sentence. We organize the words of the input sentence into a textual scene graph (TSG), which includes two kinds of edges shown in Fig.4. The black arrows indicate word-order edges, which connect words by the word order in the sentence. The brown arrows are semantic relationship edges,

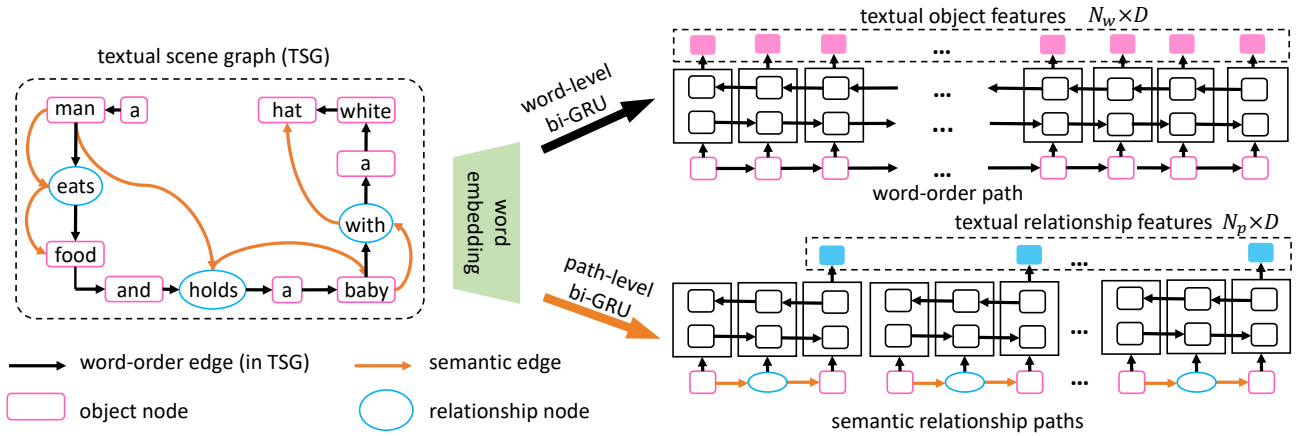

Figure 4. The architecture of the textual scene graph encoder. After embedding each word into a vector by the word embedding layer, paths connected by different edges are encoded separately by word-level bi-GRU and path-level bi-GRU.

which are built from semantic triplets parsed by SPICE [1], such as *"man-hold-baby"*. Due to different kinds of edges, different types of paths are formed in the graph. The path connected by word-order edges is named as the word-order path. Paths connected by semantic relationship edges are called semantic relationship paths.

3.2.2 Textual Scene Graph Encoder

Similar to the processing on the VSG, a textual scene graph encoder is devised to extract object and relationship features from the TSG, which consists of a word embedding layer, a word-level bi-GRU encoder, and a path-level bi-GRU encoder illustrated in Fig.4. The word-level bi-GRU encoder will encode each node along the word-order path, after which the object-level feature with context is generated at each hidden state. Due to that the semantic relationship edges break the limitation of the grammatical structure of the sentence, explicit relationship-level features are obtained after the path-level bi-GRU encodes along the semantic relationship paths.

Suppose there are N_w words and N_p semantic triplets in a sentence, its TSG will contain N_w nodes, one word-order path and N_p semantic relationship paths. Firstly, each word w_i is embedded into a vector by the word embedding layer as $e_{w_i} = W_e l_{w_i}$, where l_{w_i} is the one-hot vector of w_i and W_e is the parameter matrix of the embedding layer. We initialize W_e using the same word2vec in the VSG encoder, and then learn W_e during training end-to-end. Next, two kinds of paths are encoded separately by different bi-GRUs. For the word-order path, the word-level bi-GRU operates from the start word to the end as

$$\begin{aligned} \overrightarrow{\mathbf{h}}_{w_i} &= \overrightarrow{GRU}_w\ (\mathbf{e}_{w_i},\ \overrightarrow{\mathbf{h}}_{w_{i-1}}), \\ \overleftarrow{\mathbf{h}}_{w_i} &= \overleftarrow{GRU}_w\ (\mathbf{e}_{w_i},\ \overleftarrow{\mathbf{h}}_{w_{i+1}}), \qquad i \in [1,\ N_w], \end{aligned} \tag{4}$$

where $\overrightarrow{\mathbf{h}}_{w_i}$ and $\overleftarrow{\mathbf{h}}_{w_i}$ are the hidden vectors of w_i from two directions. Finally, the word node feature is gained as $\mathbf{h}_{w_i} = (\overrightarrow{\mathbf{h}}_{w_i} + \overleftarrow{\mathbf{h}}_{w_i})/2$, which is regarded as a textual object feature. For the N_p semantic relationship paths, each of them is encoded by the path-level bi-GRU as

$$\mathbf{h}_{p_i} = \frac{\overrightarrow{GRU}_p\ (path_i) + \overleftarrow{GRU}_p\ (path_i)}{2}, i \in [1,\ N_p] \tag{5}$$

$\mathbf{h}_{p_i}$ is the last hidden state feature of i-th semantic relationship path, which is also a relationship feature of the TSG.

3.3. Similarity Function

To measure the similarity of two encoded graphs in different modalities, we need a similarity function. Since there are two levels of features in each graph, we match them respectively. Take object features for example, let's suppose there are N_o and N_w object features in the visual and textual feature graphs, each of which is a D-dimension vector. Inspired by [11], we define the similarity score of two feature vectors h_i and h_j as $h_i^T h_j$. We calculate the similarity scores of all visual and textual object nodes, and then get a $N_w \times N_o$ score matrix, as shown in Fig.2. We find the maximum value of each row, which means for every textual object, the most related visual object among N_o visual objects is picked up. At last, we average them as the object-level score of two graphs. The relationship-level score is calculated in the same way. The above process can be formulated as

$$S^o = (\sum_{t=1}^{N_w} \max_{i \in [1,\ N_o]} \mathbf{h}_{w_t}^T \mathbf{h}_{o_i})/N_w, \tag{6}$$

$$S^r = (\sum_{t=1}^{N_p} \max_{r_{ij} \in R} \mathbf{h}_{p_t}^T \mathbf{h}_{r_{ij}})/N_p. \tag{7}$$

Finally, given a visual and textual feature graph, the similarity score is defined as $S = S^o + S^r$.

3.4. Loss Function

Triplet loss is commonly used in the image-text retrieval task, which constrains the similarity score of the matched image-text pairs larger than the similarity score of the unmatched ones by a margin, formulated as

$$L(k,l) = \sum_{\hat{l}} \max(0, m - S_{kl} + S_{k\hat{l}}) + \sum_{\hat{k}} \max(0, m - S_{kl} + S_{\hat{k}l}). \tag{8}$$

m is a margin parameter, image k and sentence l are corresponding pairs in a mini-batch, image k and sentence $\hat{l}$ are non-corresponding pairs, so are image $\hat{k}$ and sentence l. Faghri *et al*. [3] discovered that using the hardest negative in a mini-batch during training rather than all negatives samples can boost performance. Therefore, we follow [3] in this study and define the loss function as

$$L_+(k,l) = \max(0, m - S_{kl} + S_{kl'}) + \max(0, m - S_{kl} + S_{k'l}), \tag{9}$$

where $l' = \arg\max_{j \neq l} S_{kj}$ and $k' = \arg\max_{j \neq k} S_{jl}$ are the hardest negatives in the mini-batch.

4. Experiments

In the subsections, we will clarify the datasets and evaluation metrics we use for experiments. Then we give the details on implementation and show the experiment results.

4.1. Datasets and Evaluation Metrics

Flickr30k [36] and MS COCO [19] are two commonly used datasets in the image-text retrieval task, which contain $31,783$ and $123,287$ images respectively. Both of them have five text captions for each image. Following [3, 17], we split Flickr30k as $1,000$ images for validation, $1,000$ images for testing and the rest for training. For MS COCO, we split $5,000$ images for validation, $5,000$ images for testing and $113,287$ images for training.

To demonstrate the effectiveness of our approach, we conduct caption retrieval and image retrieval experiments on Flickr30k and MS COCO datasets. We adopt two universal metrics, R@k and Medr. R@k is the percentage of queries whose ground-truth is ranked within top K. Medr is the median rank of the first retrieved ground-truth.

4.2. Implementation Details

Visual Scene Graph Generation. We use Neural Motifs [37] and MSDN [18] as visual scene graph generators, which can recognize 150 categories of objects and 50 categories of relationships. We pick the top N_o (N_o=36) objects with bounding boxes and N_r (N_r=25) relationships between them sorted by classification confidence.

Table 1. Evaluation of different variants of our model on Flickr30k

model	caption retrieval			image retrieval		
	R@1	R@5	R@10	R@1	R@5	R@10
OOM **w/o** TCxt	52.7	81.8	90.3	44.6	72.2	80.9
OOM	67.6	89.7	94.5	48.6	75.6	83.8
OOM **w** VRel	65.6	89.5	95.4	50.0	77.4	85.0
OOM **w** TRel	**71.8**	91.4	**96.1**	51.0	79.5	**86.8**
SGM	**71.8**	**91.7**	95.5	**53.5**	**79.6**	86.5

Pre-trained Visual Feature Extractor. After parsing the input image into a scene graph, some objects are detected with bounding boxes. We need to transform these image regions into real-valued features. We can use the features extracted by the scene graph generator. However, most of the recent scene graph generators limit to recognize 150 categories of objects and 50 categories of relationships. Since text descriptions are rich and open-vocabulary, such visual features are not expressive enough. Moreover, most of the recent scene graph generators, including Neural Motifs [37], use VGG [23] as the backbone. In order to make a fair comparison with some state-of-the-art approaches that use Resnet [6], we prefer a feature extractor with ResNet backbone. Therefore, we take the Faster-RCNN [21] detector, which is trained on the Visual Genome dataset [16] by 1600 object classes in [2]. We use the 2048-dimension feature vector after RoI pooling.

Parameters Setting. Our SGM is implemented with Pytorch platform[1]. The output dimension of visual and textual scene graph encoder is 1024. The number of layers of GCN in visual scene graph encoder is 1. The margin m in loss function is set to 0.2. We use Adam [13] optimizer with a mini-batch size of 200 to train our model. The initial learning rate is 0.0005 for MS COCO and 0.0002 for Flickr30k.

4.3. Ablation Study

To justify the importance of relationships for image-text retrieval, we evaluate different variants of our proposed framework in Table 1. **SGM** is the full model of scene graph matching that contains relationships in both two modalities, and **OOM** is the model only considering objects matching. **OOM w VRel** and **OOM w TRel** stand for adding visual relationships and textual relationships to **OOM**, respectively. **OOM w/o TCxt** discards not only the relationships but also textual context, which means words are encoded in isolation rather than word-order bi-GRU. The best results in each column are in bold.

Impact of Relationships. From Table 1, one can find that all other models outperform **OOM w/o TCxt** that only uses isolated elements for matching. It indicates that associations between objects are essential for image-text retrieval. Comparing **SGM** with **OOM**, the importance of relationships for image-text retrieval is revealed. By adding relationship information in both modalities, the performance

[1]Our source codes are available at http://vipl.ict.ac.cn/resources/codes.

Table 2. Comparisons of state-of-the-art models on Flickr30k in cross-modal retrieval.

model	caption retrieval				image retrieval			
	R@1	R@5	R@10	Medr	R@1	R@5	R@10	Medr
VSE++ [3]	52.9	80.5	87.2	1.0	39.6	70.1	79.5	2.0
GXN [5]	56.8	-	89.6	1.0	41.5	-	80.1	2.0
SCO [8]	55.5	82.0	89.3	-	41.1	70.5	80.1	-
SCAN(t2i) AVG loss [17]	61.8	87.5	93.7	-	45.8	74.4	83.0	-
SCAN(i2t) AVG loss [17]	67.9	89.0	94.4	-	43.9	74.2	82.8	-
Ours (SGM)	**71.8**	**91.7**	**95.5**	**1.0**	**53.5**	**79.6**	**86.5**	**1.0**

Table 3. Comparisons of state-of-the-art models on MS COCO. 5k test images are the whole test dataset. 1k test images mean the test dataset is divided into five 1k subsets, and the results are the average performance on them. Results marked by '*' are our implementation with the published code and data.

model	caption retrieval				image retrieval			
	R@1	R@5	R@10	Medr	R@1	R@5	R@10	Medr
	1k Test Images							
VSE++ [3]	64.6	90.0	95.7	1.0	52.0	84.3	92.0	1.0
GXN [5]	68.5	-	**97.9**	1.0	56.6	-	94.5	1.0
SCO [8]	69.9	92.9	97.5	-	56.7	**87.5**	**94.8**	-
SCAN(t2i) AVG loss [17]	70.9	**94.5**	97.8	1.0	56.4	87.0	93.9	1.0
PVSE [25]	69.2	91.6	96.6	-	55.2	86.5	93.7	-
Ours (SGM)	**73.4**	93.8	97.8	**1.0**	**57.5**	87.3	94.3	**1.0**
	5k Test Images							
VSE++ [3]	41.3	71.1	81.2	2.0	30.3	59.4	72.4	4.0
GXN [5]	42.0	-	84.7	2.0	31.7	-	74.6	3.0
SCO [8]	42.8	72.3	83.0	-	33.1	62.9	75.5	-
*SCAN(t2i) AVG loss [17]	43.0	75.3	85.3	2.0	32.1	61.7	74.1	3.0
PVSE [25]	45.2	74.3	84.5	-	32.4	63.0	75.0	-
Ours (SGM)	**50.0**	**79.3**	**87.9**	**2.0**	**35.3**	**64.9**	**76.5**	**3.0**

has enjoyed obvious improvements (especially under the R@1) in both tasks of image retrieval and caption retrieval.

Better Representation for Retrieval. By adding visual relationships into the model, **OOM w VRel** outperforms **OOM** in image retrieval, and the same phenomenon also appears in the comparison between **SGM** and **OOM w TRel**. When considering the impact of textual relationships, similar contrasts are observed. Comparing **OOM w TRel** vs. **OOM**, and **SGM** vs. **OOM w VRel**, it shows that incorporating textual relationships is beneficial to caption retrieval. Such results suggest that better representation in one modality can make the samples in the retrieved database more differentiated and helpful to retrieval task in this modality. While for retrieval task in another modality without relationship features, gains can not be guaranteed. When we add relationship features in both modalities and match at object-level and relationship-level respectively, the performance of cross-modal retrieval obtains a large improvement.

4.4. Comparison with State-of-the-art Methods

In this section, we compare our SGM with state-of-the-art models on Flickr30k and MS COCO. For a fair comparison, all compared models use ResNet for visual feature extraction. We compared our model with VSE++ [3], GXN [5], SCO [8] and SCAN [17], which covers both global representation based model and local representation based models. VSE++ embeds full image and sentence into an embedding space and matches them. Its contribution is applying hard negatives mining in training and gaining lots of improvements. GXN leverages the image-to-text and text-to-image generative models to learn the locally grounded features. SCO concentrates on organizing semantic concepts from images into a correct order before matching with the sentence. SCAN emphasizes attending differentially to important visual objects and words by an attention mechanism. PVSE [25] addresses the issues with ambiguous instances (*e.g.* images containing multiple objects) and partial association by using K embeddings and multiple-instance learning framework. Whereas, we explore the role of relationships for image-text retrieval. The results of these methods come from their published papers or are implemented with the published code under the same evaluation protocol.

As shown in Table 2 and Table 3, our model achieves new state-of-arts on both datasets. We significantly outper-

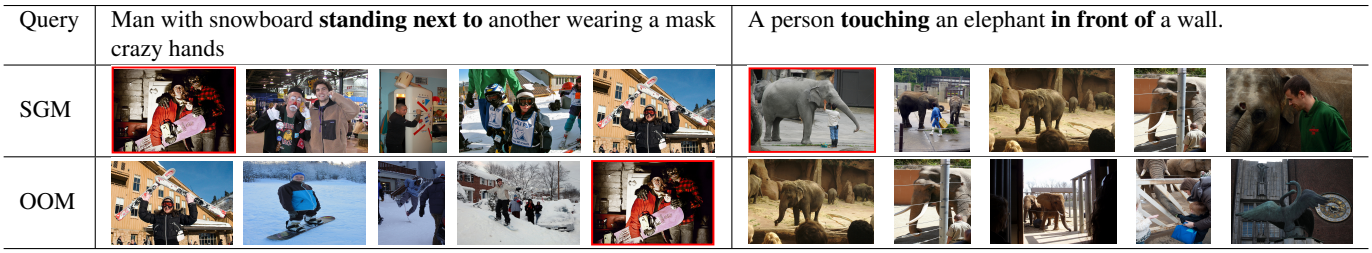

Figure 5. Qualitative top-5 image retrieval results of **SGM vs. OOM** on MS COCO. Images with red bounding boxes are the ground-truth.

Figure 6. Comparison of top-5 retrieved results before and after modifying the relationship words in queries.

form all other methods on Flickr30k and MS COCO 5k test images by a large margin. On the Flickr30k test dataset, our model outperforms the best state-of-the-art model by 16.8% relatively in caption retrieval, and 16.18% relatively in image retrieval based on R@1. On MS COCO 5k test images, we improve caption retrieval by 10.62% relatively and image retrieval by 6.65% relatively based on R@1. On MS COCO 1k test images, while our model delivers slightly lower scores than others under some metrics, it yields clearly superior performance against other competitors under the more crucial metric R@1 for retrieval task. Moreover, all local representation based models surpass the global representation based model (VSE++), which demonstrates the effectiveness of detailed matching, and the achievements of our method verify the necessity of considering relationships in image-text retrieval.

4.5. Qualitative Results

We show some image retrieval examples using **SGM** and **OOM** to reveal the importance of relationships for image-text retrieval of a complex scene. Given the same text query, the top-5 image retrieval results on MS COCO by **SGM** and **OOM** are shown in Fig.5. The top-5 retrieved images by **SGM** not only contain the right objects but also the right relationships between them. Images only contain the right objects won't be ranked at the top by **SGM**. However, results by **OOM** may overlook relationships information in queries and images. (More cases are detailed in our supplementary material.)

To prove that our **SGM** really captures relationships, we use some text queries to retrieve images from MS COCO test dataset, and then modify a relationship word in the query to retrieve again. Two retrieval results are compared in Fig.6. We can see that after modifying the relationship words in the text query, the relationships in retrieval results have changed a lot, but objects have not changed. It demonstrates our model has indeed captured the relationships so that we perform well in cross-modal retrieval task with a complex scenario. (More cases are detailed in our supplementary material.)

5. Conclusion

In this work, we proposed a graph matching based model for image-text retrieval in a complex scenario that contains various objects. We discover that not only the objects but also their relationships are important for local detailed image-text matching. To capture both objects and relationships in the images and text, we have represented image and text into the visual scene graph and the textual scene graph, respectively. Then we design the Scene Graph Matching (SGM) model to extract the object-level features and relationship-level features from the graphs by two graph encoders for image-text matching. Due to explicitly modeling relationship information, our method outperforms state-of-the-art methods in image-text retrieval experiments on both Flickr30k and MS COCO. What's more, qualitative experiments show that our approach can truly capture the relationships and is helpful in the image-text retrieval task.

Acknowledgements. This work is partially supported by Natural Science Foundation of China under contracts Nos. 61922080, U19B2036, 61772500, and CAS Frontier Science Key Research Project No. QYZDJ-SSWJSC009.

References

[1] P. Anderson, B. Fernando, M. Johnson, and S. Gould. Spice: Semantic propositional image caption evaluation. In *Proceedings of the European Conference on Computer Vision*, pages 382–398. Springer, 2016.

[2] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6077–6086, 2018.

[3] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler. Vse++: Improving visual-semantic embeddings with hard negatives. In *Proceedings of the British Machine Vision Conference*, 2018.

[4] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013.

[5] J. Gu, J. Cai, S. R. Joty, L. Niu, and G. Wang. Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7181–7189, 2018.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[7] Y. Huang, W. Wang, and L. Wang. Instance-aware image and sentence matching with selective multimodal lstm. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2310–2318, 2017.

[8] Y. Huang, Q. Wu, C. Song, and L. Wang. Learning semantic concepts and order for image and sentence matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6163–6171, 2018.

[9] J. Johnson, A. Gupta, and L. Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1219–1228, 2018.

[10] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3668–3678, 2015.

[11] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.

[12] A. Karpathy, A. Joulin, and L. Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. In *Advances in Neural Information Processing Systems*, pages 1889–1897, 2014.

[13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[14] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

[15] B. Klein, G. Lev, G. Sadeh, and L. Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4437–4446, 2015.

[16] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

[17] K.-H. Lee, X. Chen, G. Hua, H. Hu, and X. He. Stacked cross attention for image-text matching. In *Proceedings of the European Conference on Computer Vision*, pages 201–216, 2018.

[18] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang. Scene graph generation from objects, phrases and region captions. In *International Conference in Computer Vision*, pages 1261–1270, 2017.

[19] 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 *Proceedings of the European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*, 2013.

[21] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[22] 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 *Proceedings of the Fourth Workshop on Vision and Language*, pages 70–80, 2015.

[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[24] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng. Grounded compositional semantics for finding and describing images with sentences. In *Transactions of the Association for Computational Linguistics*, volume 2, pages 207–218. MIT Press, 2014.

[25] Y. Song and M. Soleymani. Polysemous visual-semantic embedding for cross-modal retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1979–1988, 2019.

[26] D. Teney, L. Liu, and A. van den Hengel. Graph-structured representations for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2017.

[27] B. Wang, Y. Yang, X. Xu, A. Hanjalic, and H. T. Shen. Adversarial cross-modal retrieval. In *Proceedings of the 25th ACM International Conference on Multimedia*, pages 154–162. ACM, 2017.

[28] L. Wang, Y. Li, and S. Lazebnik. Learning deep structure-preserving image-text embeddings. In *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition, pages 5005–5013, 2016.

[29] P. Wang, Q. Wu, J. Cao, C. Shen, L. Gao, and A. v. d. Hengel. Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1960–1968, 2019.

[30] Y.-S. Wang, C. Liu, X. Zeng, and A. Yuille. Scene graph parsing as dependency parsing. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2018.

[31] J. Wehrmann and R. C. Barros. Bidirectional retrieval made simple. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7718–7726, 2018.

[32] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*, 2019.

[33] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3097–3106, 2017.

[34] X. Yang, K. Tang, H. Zhang, and J. Cai. Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10685–10694, 2019.

[35] T. Yao, Y. Pan, Y. Li, and T. Mei. Exploring visual relationship for image captioning. In *Proceedings of the European Conference on Computer Vision*, pages 684–699, 2018.

[36] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. In *Transactions of the Association for Computational Linguistics*, volume 2, pages 67–78. MIT Press, 2014.

[37] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5831–5840, 2018.

[38] Y. Zhang and H. Lu. Deep cross-modal projection learning for image-text matching. In *Proceedings of the European Conference on Computer Vision*, pages 686–701, 2018.