A Deep Local and Global Scene-Graph Matching for Image-Text Retrieval

Manh-Duy NGUYEN [a,1], Binh T. NGUYEN [b,c,d] and Cathal GURRIN [a]

[a] *School of Computing, Dublin, Ireland*
[b] *AISIA Research Lab*
[c] *University of Science, Ho Chi Minh City, Vietnam*
[d] *Vietnam National University Ho Chi Minh City, Vietnam*

Abstract. Conventional approaches to image-text retrieval mainly focus on indexing visual objects appearing in pictures but ignore the interactions between these objects. Such objects occurrences and interactions are equivalently useful and important in this field as they are usually mentioned in the text. Scene graph presentation is a suitable method for the image-text matching challenge and obtained good results due to its ability to capture the inter-relationship information. Both images and text are represented in scene graph levels and formulate the retrieval challenge as a scene graph matching challenge. In this paper, we introduce the Local and Global Scene Graph Matching (LGSGM) model that enhances the state-of-the-art method by integrating an extra graph convolution network to capture the general information of a graph. Specifically, for a pair of scene graphs of an image and its caption, two separate models are used to learn the features of each graph's nodes and edges. Then a Siamese-structure graph convolution model is employed to embed graphs into vector forms. We finally combine the graph-level and the vector-level to calculate the similarity of this image-text pair. The empirical experiments show that our enhancement with the combination of levels can improve the performance of the baseline method by increasing the recall by more than 10% on the Flickr30k dataset. Our implementation code can be found at `https://github.com/m2man/LGSGM`.

Keywords. scene graphs, graph embedding, image retrieval

1. Introduction

Computer vision and natural language processing are two of the most popular domains of deep learning, each of which has a wide range of applications [1,2,3]. A fusion of these two areas has raised many fascinating challenges and focused the attention of researchers. One of those topics is image-text retrieval, in which a text (image) query is provided to retrieve relevant images (texts) from a given dataset. This issue's critical point is the clear semantic gap between images and text, presenting a challenging multi-modal data retrieval challenge.

[1] Corresponding Author: Manh-Duy Nguyen, School of Computing, Dublin City University, Dublin, Ireland; E-mail:manh.nguyen5@mail.dcu.ie

Many kinds of research have focused on this challenge, and one can summarise earlier works into two types. The first type includes methods [4,5,6] in which each image and textual description are entirely encoded as global representative vectors in the same space. It can be considered a decent solution due to its simplicity and fast retrieval. However, these approaches are only suitable for simple cases where there is only a single object in an image or a short sentence as they neglect the importance of positioning information. The proposed techniques [7,8,9] of the second type can deal with this positioning issue. They encode images into several regions based on detected objects. Sentences are also parsed into many fields based on their chunks of words. This approach now can gain more elaborate detail for both data. Although these methods have been shown to achieve better performance than others in the first group, there are still challenges to be solved. Neither type has considered the actions occurring within the data, including the interactions between entities/objects in images and captions. Such information can offer more in-depth insights into images as well as textual queries [10], therefore facilitating enhanced image-text retrieval. It should be mentioned that the global embedding methods in the first group also could learn this association information. However, it cannot be in detail since they do not focus directly on the relationship aspect. The introduction of a scene graph structure [10] introduces a promising way to address this issue. Due to its capacity to capture both objects and their interactions, the design has been applied and achieved excellent results in various fields [11,12,13,14]. A scene graph structure is a graph including nodes and edges connecting nodes together. In the image-text retrieval field, a node and an edge represent objects and the associations among objects detected in images or captions, as depicted in Figure 1 where the green rectangles are nodes in the graph indicating detected objects and the orange rectangles are edges illustrating the relationships between them. Recent research [15,16] could obtain better results in this retrieval area by utilizing these relations information. Hence there is a promise when applying this useful data structure in the retrieval field.

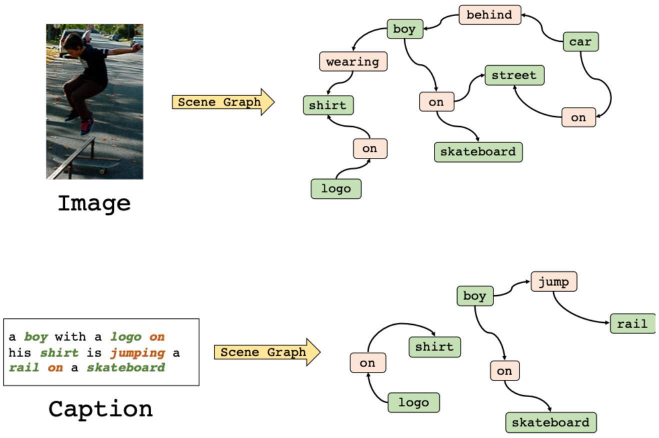

Figure 1. An example of scene graphs generated from an image and its caption. The green and the red indicates objects and their interactions respectively.

In this paper, we employ a state-of-the-art method that applies scene graph structure to solve the retrieval challenge [15]. However, we advance this baseline by introducing an addition graph convolution network to capture a global form of a graph beside its local representation produced by the baseline. To be more specific, we firstly generate scene graph information of each image and its caption. Our proposed model can extract and learn the insightful features of nodes and edges from both graphs to obtain the local details. These scene graphs are then embedded into vectors by a Siamese graph model to represent the entire image's global information and textual content. The similarity of the two graphs now can be measured at the local and global levels. We evaluate our proposed method by measuring the recall metrics on the well-known Flickr30k [17] dataset to show the effectiveness of our model.

2. Related Work

Many models have been introduced for the image-text retrieval problem. Some of them [4,5,6] encode entire images and sentences into one space to facilitate the comparison, while other solutions focused on local information instead [7,8,9]. Faghri et al. [4] followed a typical method to use a CNN-based image encoder to extract visual features of images and an RNN-based textual module to extract those from captions. These features then went through a fully connected layer to be projected into the same vector space to facilitate the comparison. Although it is a simple approach, it still achieves high performance by applying hard negatives triplet loss during training. With different points of view on analyzing sentences, Wang et al. [5] used a Bag-of-Word technique to extract the information from semantic data. However, one could also use a CNN network to process sentences in a dual-path convolutional model [6] by entirely embedding each image and sentence into a vector form using two CNN modules separately. In contrast, Huang et al. [7] focused on the local information in images where semantic concepts and their ordering were extracted by learning with their corresponding sentences. The work from Wang [9] raised the awareness of the relative position of detected objects within an image which could be useful when matching with the caption. Meanwhile, Lee et al. [8] presented a novel cross attention mechanism that learned the importance of regions (images) and words (captions) along with their alignment.

Although achieving considerable accuracy, previous work ignored or underestimated the robust information of objects' interactions within images and sentences. Johnson and colleagues firstly introduced the scene graph for dealing with this issue for the image retrieval challenge [10]. In their research, a dataset of scene graphs of images was human-annotated manually. An arbitrary graph was made and used as a query to retrieve equivalent images having a similar graph. A Conditional Random Field model and a maximum posterior technique was employed to measure the similarity of graphs. Many algorithms were invented to generate scene graphs from images [18,19,20,21]since this structure and the dataset about it was public. By virtue of them, some studies about image-text retrieval [15,16] have followed the scene graph approach recently. Shi et al. [16] created a scene concept graph based on a popular scene graph dataset [22] and used it to expand the detected concepts in images to extract visual features. Cosine similarity was used to measure the similarity between these features and the semantic features produced by applying an RNN module to captions and rank them to perform the retrieval. The scene graph matching (SGM) model from Wang et al. [15] takes a scene graph from both images and captions as input then extracts the graph features by using their own designed encoders called visual scene graph and textual scene graph for each modality. Then, one can promptly calculate the similarity between the two graphs based on each part of the first graph's agreement with every part of the second graph. Despite getting state-of-the-art results among the scene graph approaches, the SGM could still be enhanced. Firstly, its similarity calculating process only takes the small parts of a graph into account, hence neglect the overall detail. Secondly, the SGM model does not apply any normalization technique, resulting in easy overfitting. Besides, using ResNet [23] as the backbone of the structure also limits the model's potential performance. There have been several recent CNN-based networks that are more accurate than the ResNet.

Recently, some researchers have applied the attention technique to both visual and textual data to learn the interaction between the components themselves, then combine with another attention module to fuse two modalities together [24,25]. Chen et al. [24] presented a novel structure that iteratively finds matching compartments between images and sentences and refined them progressively. Meanwhile, the multi-modality cross attention network employs the compelling Transformer [26] technique on images and semantic data to get the fine-grained relationship information in detail. They both manage to get the best results so far in the domain. Nevertheless, none of them use scene graphs in their approach, which is our main focus of this research.

In this paper, we try to improve the performance of the SGM method by employing a graph convolution model to capture the overall information of a graph which can be considered as one of its weaknesses mentioned above. Our model utilizes scene graph structures for images and sentences and encodes them into vector forms for storing overall essential information to calculate the similarity between graphs instead of operating at the graph level exclusively. We name this model "Local and Global Scene Graph Matching" (LGSGM).

Our primary contributions are threefold. Firstly, the SGM measured the similarity between graphs by dividing them into sub-graphs, hence diminishing the overall detail of a graph. We overcome this problem by building a graph embedding module to summarise all information of a graph into a vector form and combine it with the graphical form to calculate the similarity of a pair of graphs. Secondly, the efficient network [27] is applied to extract the visual features of detected objects in images instead of using conventional Residual structure [23] in the SGM model as the former network is shown to be more accurate although having a simpler design. Moreover, we also integrate normalization techniques including Dropout and Batch normalization to mitigate the potential of overfitting. Thirdly, we run an evaluation on the Flickr30k dataset for the image-text matching problem to facilitate a comparison with the SGM model [15] as the baseline, which also uses scene graphs to support retrieval. We choose the SGM model as the baseline due to its state-of-the-art result in this field for models that employ scene graphs in the structure.

3. Methodology

This section will describe how our LGSGM model approaches the retrieval challenge as a graph similarity ranking problem which is inspired by the SGM. Figure 2 depicts the workflow of LGSGM that integrates an extra graph embedding stage to the SGM baseline. Initially, our preprocessing data stage starts with the scene graph construction.

The graphs of all images and texts are extracted beforehand and stored in the database. A generated scene graph of a query, which can be an image or a sentence, then goes through a graph encoder module to get a graph feature. Meanwhile, each sample in the remaining modality database also follows the same scheme but separately. All feature graphs now are passed through a Siamese-structured [28] graph embedding layer to learn the summarised attributes of them, and the vector-level features of those graphs are obtained. The similarity of a pair of graphs, which is a query graph and a sample graph in the database, now can be calculated based on the feature graphs and their embedded vectors representing local and global information accordingly. Finally, these scores are ranked descending to find the most relevant answers from the database.

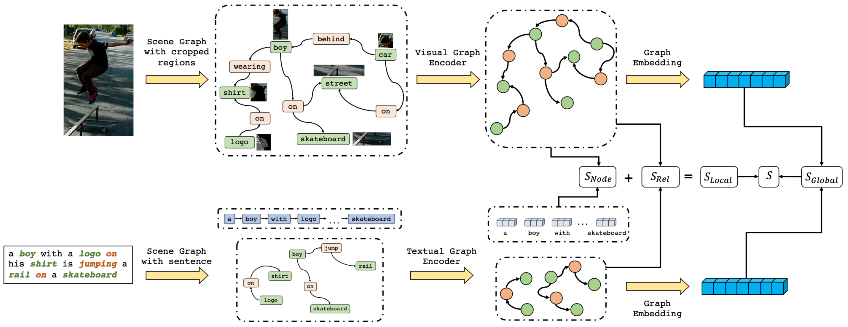

Figure 2. Our proposed LGSGM pipeline follows the SGM baseline. Scene graphs of an image and a query are firstly extracted at the preprocessing stage. Both graphs are encoded by different graph encoders and embedded into vector-level forms by a share graph embedding network to get to local and global features, which will be used to calculate the similarity between them.

3.1. Visual Graph Encoder

A scene graph of an image can be established beforehand by using any available scene graph generation method. The graph G after extracted can be seen as a tuple $G = (O, B, R)$ where:

- $O = \{o_1, ..., o_{N_o}\}$ is the set of N_o semantic labels of detected objects appearing the image.
- $B = \{b_{o_1}, ..., b_{o_{N_o}} | b_{o_i} \in \mathbb{R}^4\}$ is the set of bounding boxes where b_{o_i} is the coordinate of the box of the corresponding object o_i in the set O.
- $R = \{r_1, ..., r_{N_r}\}$ is the set of N_r predicted relations between objects in the image. Each r_m is a tuple of (o_i, p_{ij}, o_j) where p_{ij} is the label of the association between two objects o_i and o_j. It is noted that R can be seen as the set of edges connecting nodes, which are objects in O, of the scene graph G.

As shown in Figure 3, after the graph is constructed, we embed the label of the nodes and edges into vectors with the help of trainable word embedding layers to gain their semantic information. Besides, there is also rich information from images themselves. We extract an image feature for each object in O based on its associated bounding box in B. Regarding an edge p_{ij}, its image feature can be computed through the union regions of two boxes b_{o_i} and b_{o_j}. These regions are fed into a CNN-based model, which is the

EfficientNet [27] in our setting, to get the image features. The feature of one node or an edge now is the fusion of its image feature and semantic features. We update these features to learn the connection between nodes and edges features by using a convolutional graph neural network [29] to get the final visual feature graph. The graph is finally fed to the Siamese graph embedding model to get its global representation.

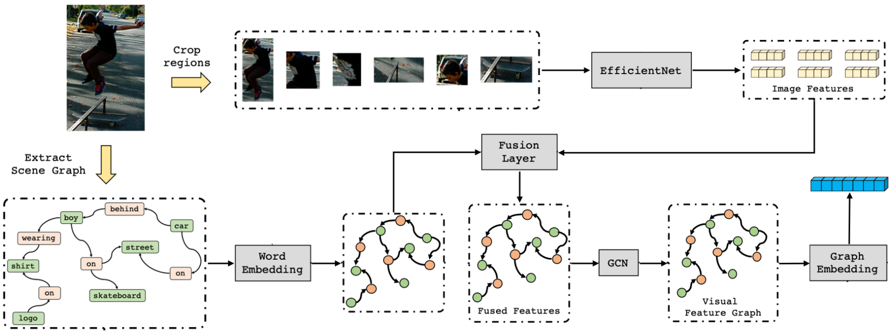

Figure 3. The workflow of an image in our LGSGM model.

Word Embedding. We exploit the information of semantic labels by embedding names of objects and relations in O and R into vectors with two distinct embedding layers for each. Each object label o_i and predicate label p_{ij} is transformed into one-hot vector I_{o_i} and $I_{p_{ij}}$. Their embedding vectors, e_{o_i} and $e_{p_{ij}}$, can be calculated as

$$e_{o_i} = W_o I_{o_i} \ , W_o \in \mathbb{R}^{d_W \times C_o} \tag{1}$$

$$e_{p_{ij}} = W_p I_{p_{ij}} \ , W_p \in \mathbb{R}^{d_W \times C_p} \tag{2}$$

where W_o and W_p are the trainable parameters in the layer, C_o and C_p is the number of categories of objects and relations supported in the scene graph generation method. We set $d_W = 300$ and utilise the pretrained embedding Glove model to initialise W_o and W_p.

Image Features. An image feature v_{o_i} of an object o_i can be obtained by applying a pretrained CNN-based network to the cropped region b_{o_i} around the object. Similarly, the cropped region for an edge p_{ij}, which is the area covering both b_{o_i} and b_{o_j} regions, also be used to extract its d_I-dimension image feature v_{ij}.

Fused Features. A simple trainable, fully connected neural network is applied to the semantic feature's concatenated vector and the image feature to get the fused feature that can combine both modality detail. The fused representation of a node and an edge is achieved as followed:

$$u_{o_i} = f_{act}(W_u[v_{o_i}, e_{o_i}]) \tag{3}$$

$$u_{ij} = f_{act}(W_u[v_{ij}, e_{ij}]), \tag{4}$$

where $[.]$ is the concatenating operation, f_{act} is an activation function, and $W_u \in \mathbb{R}^{d_F \times (d_I + d_W)}$ is the trainable parameters.

Graph Network. We use a Convolutional Graph Network (GCN) to learn the connection and update the fused features of nodes (u_{o_i}) and edges (u_{ij}) of the graph. The GCN works similarly to a normal CNN operation but more flexible to many types of graph-structured data than only the grid data of a conventional CNN. In this GCN model, the features of a node are only updated based on itself solely to mitigate noise effects from surrounding nodes [15]. In contrast, because an edge bridges two nodes together, its features should be related to those nodes' features to learn the association between them. Regarding an n-layer GCN model, updated features of a node and an edge at the l^{th} layer can be formulated as follows:

$$h^l_{o_i} = \mathrm{MLP}_o(h^{l-1}_{o_i}) \tag{5}$$

$$h^l_{p_{ij}} = \mathrm{MLP}_p([h^{l-1}_{o_i}, h^{l-1}_{p_{ij}}, h^{l-1}_{o_j}]), \tag{6}$$

where MLP_o and MLP_p are two separate neural network models, and $h^0_{o_i} = u_{o_i}$ and $h^0_{p_{ij}} = u_{ij}$. The final output of the visual feature graph is the updated features of nodes and edges which are denoted as h_{o_i} and $h_{p_{ij}}$.

3.2. Textual Graph Encoder

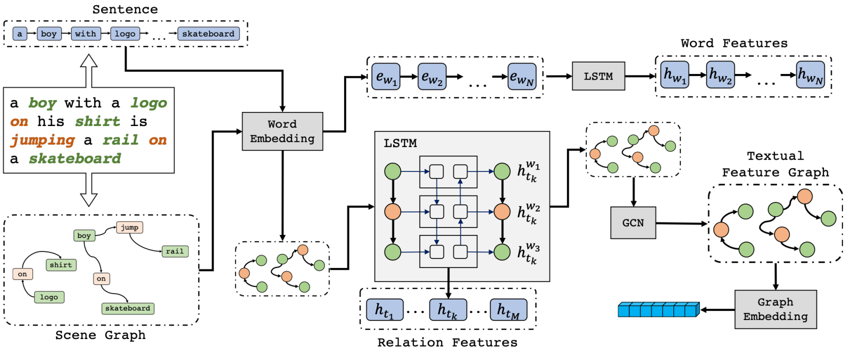

Figure 4. The workflow of a sentence in our LGSGM model. The *N* and *M* in the figure indicate the number of words and number of relations in the sentence.

Similarly, a sentence also follows the same process of images. Figure 4 illustrates the process of getting local and global features of the sentence. Firstly, the description is converted into a scene graph to describe the relationships in detail. These triplet information, such as *"man-wears-hat"*, can be obtained by using SPICE technique [13]. However, relying on the graph only will ignore other important clues in the sentence. Therefore two different modules are used to extract the data from both pathways. Initially, every word in the sentence can be encoded by a word embedding module similar to that in the visual feature graph. Two distinct LSTM models [30] are then employed to learn the features of each word in the sentence as well as the features of each extracted relation triplet. We expand the SGM model by introducing an additional GCN to the graph created from the triplets. This graph is used to summarise information of all relations in the sentence and compared to that of visual modality afterward.

Word Embedding. It is worth noting that there are a massive number of vocabularies in a real-world deployment. Hence, it is not suitable for the word embedding part to learn all words in the entire semantic modality database as an overfitting issue will occur in the test condition. We create the dictionary from the textual database (excluding the test set) and only learn some popular words. We replace the least common ones, whose appearance frequency less than 4, with the *"unknown"* word, which is considered a popular vocabulary. During the test phase, words not included in the dictionary are converted to *"unknown"*.

Word Features. After the embedding stage, the entire sentence goes to the bidirectional LSTM hierarchy to get the features from both forwards and backward ways. Then the representation of a word w_i in the sentence, denoted as h_{w_i}, is the average of two paths and computed as follows:

$$h_{w_i} = \frac{\overrightarrow{\text{LSTM}_w}(e_{w_i}, \overrightarrow{h_{w_{i-1}}}) + \overleftarrow{\text{LSTM}_w}(e_{w_i}, \overleftarrow{h_{w_{i+1}}})}{2}, \tag{7}$$

where LSTM$_w$ is the bi-LSTM model, e_{w_i} is the word embedding vector whilst $\overrightarrow{h_{w_{i-1}}}$ and $\overleftarrow{h_{w_{i+1}}}$ are the hidden states of the word w_i from forward and backward directions.

Graph Network. We organise the set of relations of a caption as $T = \{t_1, ..., t_{N_t}\}$ with N_t is the total number of relations in the sentence parsed by SPICE. Each triplet t_m is a sequence of words (w_i, w_{ij}, w_j) where the word w_{ij} is the association between two objects w_i and w_j such as ("man", "wears", "hat"). We use another bidirectional LSTM structure, named LSTM$_t$, and share it among all triplet t_i for $i \in [1, N_t]$. The feature of a word in a triplet is the hidden state itself while the feature of entire triplet is the last state of the sequence model. Specifically, they are calculated as:

$$h_{t_k}^{w_n} = \frac{\overrightarrow{\text{LSTM}_t}(e_{t_k}^{w_n}, \overrightarrow{h_{t_k}^{w_{n-1}}}) + \overleftarrow{\text{LSTM}_t}(e_{t_k}^{w_n}, \overleftarrow{h_{t_k}^{w_{n+1}}})}{2}, \tag{8}$$

$$h_{t_k} = \frac{\overrightarrow{h_{t_k}^{w_i}} + \overleftarrow{h_{t_k}^{w_j}}}{2}, \tag{9}$$

where triplet t_k is a sequence of words started with w_i and ended with w_j, $e_{t_k}^{w_n}$ is the word embedding vector of w_n in t_k. An extra GCN model having a similar structure with that in visual modality (Eq. 5 and Eq. 6) is then applied to the graph formed by the set T. Particularly, the graph will have $h_{t_k}^{w_n}$ as feature of nodes and $h_{t_k}^{w_{ij}}$ for edges which are all go through two neural networks to get the graph-level features of nodes and edges which are denoted as $hg_{t_k}^{w_n}$ and $hg_{t_k}^{w_{ij}}$.

3.3. Graph Embedding

After feature graphs of both modalities are extracted, we embed them into vector forms using an attention mechanism called multi-scale node attention [31]. Given a graph G

having N nodes $\{h_1,...,h_N\}$ and M edges $\{r_1,...,r_M\}$ where h_i and $r_j \in \mathbb{R}^d$ is a feature of a node and an edge accordingly, the embedded vector of G, noted as $a_G \in \mathbb{R}^{2d}$, is obtained by the following formula:

$$a_G^h = \sum_{n=1}^{N} \sigma(h_n^{\mathrm{T}} \mathrm{ReLU}(W_h(\frac{1}{N}\sum_{m=1}^{N} h_m)))h_n, \tag{10}$$

$$a_G^r = \sum_{n=1}^{M} \sigma(r_n^{\mathrm{T}} \mathrm{ReLU}(W_r(\frac{1}{M}\sum_{m=1}^{M} r_m)))r_n, \tag{11}$$

$$a_G = [a_G^h, a_G^r] \tag{12}$$

where σ and ReLU indicate the sigmoid and rectified linear unit activate function, W_h and $W_r \in \mathbb{R}^{d\times d}$ is the training parameter in the model, subscript T denotes the transpose operation, and [.] is the concatenation. We share this attention structure to the visual feature graph and textual feature graph as a siamese-model to get their vector-level representation a_V and a_T respectively. It is noted that the node features and edge features of a visual graph are h_{o_i} and $h_{p_{ij}}$ while those of textual graph are $hg_{t_k}^{w_n}$ and $hg_{t_k}^{w_{ij}}$.

3.4. Similarity Function

The similarity between two scene graphs is measured using both local and global approaches. In the local form, each part of a graph is compared to every part of other graphs; for instance, matching on each node-level feature and on each edge-level feature. Regarding the global approach, the embedding vectors of the entire two graphs are used. This similarity score can be used to rank the retrieved result as higher means more relevant answers.

Local similarity. The local score is the sum of a node score and an edge score. The former score is the average of the matching scores of all words in a sentence with their most relevant object detected in an image. The relevant score of a word w_i in the caption with an object in the image o_k is their dot product $h_{w_i}^{\mathrm{T}} h_{o_k}$. Assuming there are N_w words in the caption and N_o objects in the image, the node score is calculated as:

$$S_{Node} = \frac{1}{N_w}\left(\sum_{i=1}^{N_w} \max_{k\in[1,N_o]} h_{w_i}^{\mathrm{T}} h_{o_k}\right) \tag{13}$$

Similarly, the edge score is the mean score between relations in the textual data with their most matching edge in the visual graph and is formulated as:

$$S_{Rel} = \frac{1}{N_t}\left(\sum_{i=1}^{N_t} \max_{p_{ij}\in R} h_{t_i}^{\mathrm{T}} h_{p_{ij}}\right) \tag{14}$$

The local similarity then can be obtained as follows:

$$S_{Local} = S_{Node} + S_{Rel} \tag{15}$$

Global similarity. The global vectors provide the overall information of graphs. We use conventional cosine distance to measure the degree of matching between two entire graphs. Specifically, the similarity score, S_{Global}, of two embedding vectors of visual graph a_V and textual graph a_T is:

$$S_{Global} = \frac{a_V^{\mathrm{T}} a_T}{\|a_V\| \, \|a_T\|} \tag{16}$$

Finally, the similarity between two graphs takes both local and global details into account, hence is calculated by the sum of them, which is

$$S = S_{Local} + S_{Global} \tag{17}$$

This bi-level similarity score is also used in the loss function during the training phase where we apply the hardest negative triplet loss as in [4,15]. The loss function is:

$$L(k,l) = \max(0, m - S_{kl} + S_{k\hat{l}}) + \max(0, m - S_{kl} + S_{\hat{k}l}), \tag{18}$$

where m is a margin hyperparameter, S_{kl} is the similarity score of a true pair of image k and its caption l, and $\hat{k}$, $\hat{l}$ is the least matching image and sentence with l and k in the mini-batch, respectively.

4. Dataset and Metrics

We evaluate our proposed model on the Flickr30k dataset [17], which is one of the most popular datasets in this image-text matching field due to its high quality of textual annotation compared to others [32]. The dataset consists of $31,783$ images, and each of them has five corresponding captions. An example of image-sentence pairs and their generated scene graph can be illustrated in Figure 1. We split the Flickr30k data into training, validating, and testing sets, so that the number of the training images is $29,783$, whilst the two latter subsets both have $1,000$ images.

The evaluation metric we choose is a common Recall at K (R@K) value that has been used in many kinds of research [15,9,24,25]. The R@K is the proportion of queries that we find their correct matching answers in the top K of the ranking result. In our experiments, we evaluate three values of K, which are 1, 5, and 10.

5. Experiments and Results

In this section, we run our proposed model on the Flickr30k dataset then compare the baseline that is chosen as the SGM model as both our LGSGM and the SGM have a similar scene graph approach.

5.1. Model Setting

At the scene graph generation stage, we use Neural Motifs [19] to get the similar graphs as described in [15]. We only keep the top $N_o = 36$ detected objects and $N_r = 25$ predicates in the result sorted by their confidence scores for each scene graph. Regarding visual features of objects, we employ the EfficientNet, which is considered to be more accurate in the image classification challenge, although fewer parameters [27] than ResNet [23] as used in the baseline. The region surrounding the object is firstly cropped based on its bounding box and resized to the desired format, then goes through the net, which is excluded from the classification layer, to get the feature. We set $d_I = d_F = 2048$ and choose the EfficientNet-b5 model pretrained on ImageNet to get the 2048-dim feature vectors as comparable with the baseline. The number of layers in GCN models for both modalities and the LSTM models is 1. The dimension of the output vector from GCN is configured to be $d = 1024$, which is also the size of the hidden state of LSTM models. We also apply Dropout and Batch-normalisation to avoid the potential overfitting in both hierarchies. We use Swish [33] as an activate function when creating fused features (Eq. 3 and Eq. 4) and in GCN models. We also tried other activation functions (such as ReLU, Leaky ReLU, or Tanh), but Swish has the best performance among these functions. The margin m in the loss function (Eq. 18) is selected as 0.35. We train the model with the batch size of 128 at the learning rate of 0.0003 with Adam optimizer.

5.2. Comparison with Baseline and other Methods

Although our primary focus is to compare with the baseline SGM, we also show the result of other state-of-the-art techniques to provide a comprehensive perspective on the performance of our proposed approach. Besides the SGM, other competing models are:

- PFAN [9] which uses the position focused attention network to extract the features of the location of objects in images.
- IMRAM [24] with the attention mechanism to learn the matching fragments between images and sentences.
- MMCA [25] applying attention and transformer compartment to exploit the relationship between objects in images and words in texts within themselves.
- GSMN [34] that is also a graph-based approach but connects all detected objects within an image to create a graph then compares to that of a text.

The LGSGM and others' results on Flickr30k can be depicted in Table 1. The value in bold is the highest number in that metric. Caption retrieval indicates that a model needs to find texts that are relevant to a query image. On the contrary, image retrieval is used when a query is a sentence. It is important to note that other models' metrics are taken from their original report since we use the same subset for training, validating, and testing.

Table 1. Performance of models on Flickr30k Dataset. R-Sum is the sum of all recall metrics.

Methods	**Caption retrieval**				**Image retrieval**			
	R@1	R@5	R@10	R-Sum	R@1	R@5	R@10	R-Sum
PFAN [9]	70	91.8	95.0	256.8	50.4	78.7	86.1	215.2
IMRAM [24]	74.1	93.0	96.6	263.7	53.9	79.4	87.2	220.5
MMCA [25]	74.2	92.8	96.4	263.4	54.8	81.4	87.8	224.0
GSMN [34]	**76.4**	**94.3**	**97.3**	**268.0**	**57.4**	82.3	89.0	228.7
SGM [15]	71.8	91.7	95.5	259.0	53.5	79.6	86.5	219.6
LGSGM (Ours)	71	91.9	96.1	259.0	**57.4**	**84.1**	**90.2**	**231.7**

It is easier for all models to find captions when an image is given as a query than the reversed retrieving because the caption retrieval section's scores are higher than those in image retrieval. Regarding the caption retrieval, the GSMN using an ensemble setting performs best among all models, where the differences in the R-Sum can be up to more than 10%. This model is more accurate than ours with 3% on average of recall. Both attention-based techniques, IMRAM and MMCA, share similar results as there is no significant gap in all recall metrics. It is also true for SGM and LGSGM models. Although we obtain higher scores at R@5 and R@10 with a margin less than 0.5%, the SGM model has a slightly better R@1 metric than ours, which are 71.8% and 71%, respectively. The PFAN model performs worst with the lowest R@1 and R@10. However, our LGSGM achieves the highest scores on all three recall metrics in the image retrieval field, which are 57.4%, 84.1%, and 90.2% accordingly. It makes the proposed method have the best R-Sum of 231.7% in this text-to-image retrieval, which is 3% higher than the GSMN in the second place. In specific, our R@5 and R@10 are better than that of GSMN roughly by 1.8% and 1.2% respectively. Compared with the SGM, our improvement creates a huge increase of 12.1% in total recall. The PFAN is still the model with the lowest recall, while the R-Sum of MMCA is 4.5% higher than IMRAM.

In general, the proposed LGSGM surpasses the SGM model by a large margin, which is also our main contribution. Our model and GSMN, which is also a graph-based method, are the top-2 methods in the experiment, showing the usefulness of the graph structure in this field. Nevertheless, GSMN achieves higher recall than ours with 496.7% compared to 490.7% of our model in the total of all recall values in both types of retrieval. It might be due to its dense graph structure where this model connects all of the objects in an image while the scene graph structure only captures some detected relations between them. Although the attention approaches are better than our scene graph model in the image-to-text retrieval, our network still manages to get the highest score on the remaining experiment. Moreover, LGSGM manages to score better than those models concerning the sum of recall of both image retrieval and caption retrieval experiments. With our state-of-the-art result in the text-to-image retrieval section, it opens a wide range of applications of our structure in the field of finding images that are relevant to the given description. For instance, one potential application is in lifelogging retrieval, where a graph-based method has shown its promising performance [14].

6. Conclusion

In this research, we address an issue that remained in the state-of-the-art scene graph matching model in which the global detail of graphs is ignored during the graph encoding phrase. We propose a graph embedding module that can address that concern by summarising the overall information of a graph into a vector form. Our LGSGM method, therefore, can measure the similarity between images and captions based on their input scene graphs with both local and global views. Using a lighter and more accurate EfficientNet to extract features combining with normalizing techniques to mitigate the overfitting problem, our model can surpass the baseline and achieve the new state-of-the-art results for those using scene graph as input in image-text retrieval challenge.

7. Acknowledgments

This publication has emanated from research supported in party by research grants from Science Foundation Ireland under grant numbers SFI/12/RC/2289, SFI/13/RC/2106, and 18/CRT/6223.

References

[1] Duy NHM, Duy NM, Truong MTN, Bao PT, Binh NT. Accurate brain extraction using Active Shape Model and Convolutional Neural Networks. arXiv preprint arXiv:180201268. 2018.

[2] Shih ML, Su SY, Kopf J, Huang JB. 3D Photography using Context-aware Layered Depth Inpainting. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2020. .

[3] Johnson M, Schuster M, Le QV, Krikun M, Wu Y, Chen Z, et al. Google's multilingual neural machine translation system: Enabling zero-shot translation. Transactions of the Association for Computational Linguistics. 2017;5:339–351.

[4] Faghri F, Fleet DJ, Kiros JR, Fidler S. Vse++: Improving visual-semantic embeddings with hard negatives. arXiv preprint arXiv:170705612. 2017.

[5] Wang B, Yang Y, Xu X, Hanjalic A, Shen HT. Adversarial cross-modal retrieval. In: Proceedings of the 25th ACM international conference on Multimedia; 2017. p. 154–162.

[6] Zheng Z, Zheng L, Garrett M, Yang Y, Xu M, Shen YD. Dual-path convolutional image-text embeddings with instance loss. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM). 2020;16(2):1–23.

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

[8] Lee KH, Chen X, Hua G, Hu H, He X. Stacked cross attention for image-text matching. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. p. 201–216.

[9] Wang Y, Yang H, Qian X, Ma L, Lu J, Li B, et al. Position focused attention network for image-text matching. arXiv preprint arXiv:190709748. 2019.

[10] Johnson J, Krishna R, Stark M, Li LJ, Shamma D, Bernstein M, et al. Image retrieval using scene graphs. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. p. 3668–3678.

[11] Johnson J, Gupta A, Fei-Fei L. Image generation from scene graphs. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 1219–1228.

[12] Xu N, Liu AA, Liu J, Nie W, Su Y. Scene graph captioner: Image captioning based on structural visual representation. Journal of Visual Communication and Image Representation. 2019;58:477–485.

[13] Anderson P, Fernando B, Johnson M, Gould S. Spice: Semantic propositional image caption evaluation. In: European conference on computer vision. Springer; 2016. p. 382–398.

[14] Nguyen MD, Nguyen BT, Gurrin C. Graph-Based Indexing and Retrieval of Lifelog Data. In: International Conference on Multimedia Modeling. Springer; 2021. p. 256–267.

[15] Wang S, Wang R, Yao Z, Shan S, Chen X. Cross-modal scene graph matching for relationship-aware image-text retrieval. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision; 2020. p. 1508–1517.

[16] Shi B, Ji L, Lu P, Niu Z, Duan N. Knowledge Aware Semantic Concept Expansion for Image-Text Matching. In: IJCAI. vol. 1; 2019. p. 2.

[17] Young P, Lai A, Hodosh M, Hockenmaier J. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. Transactions of the Association for Computational Linguistics. 2014;2:67–78.

[18] Yang J, Lu J, Lee S, Batra D, Parikh D. Graph r-cnn for scene graph generation. In: Proceedings of the European conference on computer vision (ECCV); 2018. p. 670–685.

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

[20] Tang K, Zhang H, Wu B, Luo W, Liu W. Learning to compose dynamic tree structures for visual contexts. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. p. 6619–6628.

[21] Li Y, Ouyang W, Zhou B, Shi J, Zhang C, Wang X. Factorizable net: an efficient subgraph-based framework for scene graph generation. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. p. 335–351.

[22] Krishna R, Zhu Y, Groth O, Johnson J, Hata K, Kravitz J, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. International journal of computer vision. 2017;123(1):32–73.

[23] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 770–778.

[24] Chen H, Ding G, Liu X, Lin Z, Liu J, Han J. Imram: Iterative matching with recurrent attention memory for cross-modal image-text retrieval. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. p. 12655–12663.

[25] Wei X, Zhang T, Li Y, Zhang Y, Wu F. Multi-modality cross attention network for image and sentence matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. p. 10941–10950.

[26] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. arXiv preprint arXiv:170603762. 2017.

[27] Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning. PMLR; 2019. p. 6105–6114.

[28] Koch G, Zemel R, Salakhutdinov R. Siamese neural networks for one-shot image recognition. In: ICML deep learning workshop. vol. 2. Lille; 2015. .

[29] Wu Z, Pan S, Chen F, Long G, Zhang C, Philip SY. A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems. 2020.

[30] Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997;9(8):1735–1780.

[31] Bai Y, Ding H, Qiao Y, Marinovic A, Gu K, Chen T, et al. Unsupervised inductive graph-level representation learning via graph-graph proximity. arXiv preprint arXiv:190401098. 2019.

[32] Plummer BA, Wang L, Cervantes CM, Caicedo JC, Hockenmaier J, Lazebnik S. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In: Proceedings of the IEEE international conference on computer vision; 2015. p. 2641–2649.

[33] Ramachandran P, Zoph B, Le QV. Searching for activation functions. arXiv preprint arXiv:171005941. 2017.

[34] Liu C, Mao Z, Zhang T, Xie H, Wang B, Zhang Y. Graph structured network for image-text matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. p. 10921–10930.