Learning Fragment Self-Attention Embeddings for Image-Text Matching

Yiling Wu[1,2,3], Shuhui Wang[1,*], Guoli Song[1,2,3], Qingming Huang[1,2,3]
[1] Key Lab of Intell. Info. Process., Inst. of Comput. Tech., Chinese Academy of Sciences, Beijing, 100190, China.
[2] School of Computer Sci. and Tech., University of Chinese Academy of Sciences, Beijing, 101408, China.
[3] Key Lab of Big Data Mining and Knowledge Management, University of Chinese Academy of Sciences, Beijing, 100190, China.
yiling.wu@vipl.ict.ac.cn,wangshuhui@ict.ac.cn,guoli.song@vipl.ict.ac.cn,qmhuang@ucas.ac.cn

ABSTRACT

In image-text matching task, the key to good matching quality is to capture the rich contextual dependencies between fragments of image and text. However, previous works either simply aggregate the similarity of all possible pairs of image regions and words, or take multi-step cross attention to attend to image regions and words with each other as context, which requires exhaustive similarity computation between all image region and word pairs. In this paper, we propose Self-Attention Embeddings (SAEM) to exploit fragment relations in images or texts by self-attention mechanism, and aggregate fragment information into visual and textual embeddings. Specifically, SAEM extracts salient image regions based on bottom-up attention, and takes WordPiece tokens as sentence fragments. The self-attention layers are built to model subtle and fine-grained fragment relation in image and text respectively, which consists of multi-head self-attention sub-layer and position-wise feed-forward network sub-layer. Consequently, the fragment self-attention mechanism can discover the fragment relations and identify the semantically salient regions in images or words in sentences, and capture their interaction more accurately. By simultaneously exploiting the fine-grained fragment relation in both visual and textual modalities, our method produces more semantically consistent embeddings for representing images and texts, and demonstrates promising image-text matching accuracy and high efficiency on Flickr30K and MSCOCO datasets.

CCS CONCEPTS

• **Computing methodologies** → **Learning latent representations**; *Neural networks*; • **Information systems** → *Information retrieval*.

KEYWORDS

image-text matching, self-attention, fragment embeddings

ACM Reference Format:
Yiling Wu, Shuhui Wang, Guoli Song, Qingming Huang. 2019. Learning Fragment Self-Attention Embeddings for Image-Text Matching. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), October 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3343031.3350940

1 INTRODUCTION

Cross-modal retrieval (CMR) [31] has attracted much more attention in multimedia research in recent years. Given queries in one modality, CMR aims to retrieve documents from another modality. Among various cross-modal retrieval tasks, image-text matching is one of the most important tasks which involves visual and linguistic understanding, which enables users to find images that best illustrate the topic of textual query, or textual descriptions that best explain the content of visual query. However, data in different modalities are represented in heterogeneous feature spaces, and thus they have distinguished statistical properties. The modality heterogeneity leads to great challenge in measuring the semantic relevance among massive cross-modal data objects.

There has been a surge of research interests in tackling the challenging image-text matching problem. The simplest way is to learn a pair of linear projection functions to map visual and textual data into a unified latent space [8, 31]. Driven by the success of deep learning, the main stream has been changed to modality-specific deep feature learning, *e.g.*, learning CNN for image and RNN for text. Most existing solutions are towards learning global representations for image and sentence by enforcing constraints encoded as triplet ranking loss [38] or correlation maximization [3, 46]. However, for global feature representation, the important parts, *i.e.*, the image regions delivering salient semantics, cannot be well focused on. For example, it has been widely recognized that objects in an image tend to be more semantically correlated to its matched sentence, while the visual background contains less information in describing semantic image-text correlation. Some other works [10, 25, 39, 45] use feature representation from the last pooling layer of CNNs to preserve the spatial information of the original image, thus the pixel-level attention [40, 42] or co-attention can be learned for consequent processing. These approaches cannot guarantee to find all the salient pixels containing meaningful information, and suffer from over-fitting.

*Corresponding author.
Codes are available at https://github.com/yiling2018/saem.

MM '19, October 21–25, 2019, Nice, France

ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3350940

We first address the issue of how to extract fine-grained semantically salient image patches and words in sentences for modeling image-text relation. Specifically, owning to the success of object detection [32], we use salient regions in the image at task-independent object/stuff level in a way analogous to the spontaneous bottom-up attention associated with unexpected, novel or salient stimuli in human vision system [2]. We use WordPiece tokens of each sentence as the fragment in textual modality. The visual and textual fragments should be correctly organized to model the image-text matching relation. One simple way is to utilize the aggregate similarity of all fragments of image and text [12]. Cross attention or co-attention which involves multi-step of attending to image regions based on text or attending to words based on image [17, 18] can also be applied. However, existing strategies require computational demanding pairwise similarity computation between all image-text pairs with complex methods at test stage, which lack efficiency in real-world application scenarios. Instead of exhaustively computing similarities of all pairs of image regions and words in sentence, we consider learning embeddings for images and texts which independently project the two heterogeneous data modalities into a joint space. Thus, similarity between image and text can be directed compared on the learned embeddings.

Considering that regions in an image are usually correlated, *e.g.*, plates usually co-occur with tables in an indoor scene, we discover the relations between image object regions based on self-attention mechanism [36]. Specifically, we build a self-attention layer to calculate the similarities between all fragments and compute a weighted fragment combination according to the central query fragment. With self-attention mechanism, the distance between each fragment is one constantly. Different from previous works [27, 48] using LSTM to organize image fragments, the fragments do not need to be organized into a linear sequence, which can avoid considering the complex but ambiguous long-range/high-order fragment dependencies. Since self-attention mechanism lets each fragment see other fragments in the same image, we get a set of region-centered image embeddings after the self-attention layer. Simple average pooling operation is then used to aggregate the fragment representations, which resembles the bag-of-visual-words model.

For textual modality, we adopt the well pre-trained BERT [4] which consists of multiple self-attention layers to extract a set of context sensitive word representations. Different from Word2Vec [24] and GloVe [28] which give the same embedding to a word without considering the context, the word embeddings given by BERT encode semantic context more accurately. After getting word representations, 1d-convolution neural network is adopted to exploit phrase-level information, and fully connected layer is then applied to get the global embedding of text. Based on the visual and textual fragment self-attention, we use inner product to calculate the similarity of image embeddings and text embeddings, so that performing average pooling on region-centered image embeddings is equivalent to aggregating the similarities between multiple region-centered image embeddings and a text embedding.

To learn a joint low-dimensional embedding space for image and text, we minimize a combination of bi-directional triplet loss [43, 44] and bi-directional angular loss [37] with hard negative mining for training. Since images and sentences are independently embedded into the joint space without being paired as in co-attention [39] or cross-attention [17], efficient retrieval can be performed at test stage. To evaluate the performance of our approach in comparison to other architectures, we conduct experiments on Flickr30K dataset [47], and MSCOCO dataset [20]. Promising results have been achieved by our approach on image-sentence retrieval tasks on MSCOCO and Flickr30K datasets, which demonstrates the remarkable accuracy and high efficiency of our SAEM in embedding sentences and images for image-sentence retrieval.

2 RELATED WORK

2.1 Cross-modal Matching

There are mainly two paradigms of cross-modal matching methods: 1) embedding learning [1, 6, 30, 38, 46, 50] and 2) pairwise similarity learning [17, 22, 23, 39, 43].

Embedding learning paradigm focuses on embedding images and texts into a latent space so that they can be compared directly using simple distance metric. The canonical correlation analysis (CCA) [8] learns a latent space by maximizing the correlation between the projected data of two modalities. As labels contain rich semantic information, CCA is extended by using label information [1, 6, 30], constructing non-parametric mappings [35] and non-linear cross-modal projections [9]. With the achievement in deep learning, the performance of cross-modal retrieval has been significantly improved in recent years. Kiros *et al.* [15] use CNNs to encode images and RNNs to encode sentences, and learn image and text embeddings with a hinge-based triplet ranking loss for bi-directional ranking. Wang *et al.* [38] propose to match images and texts using hinge-based triplet ranking objective that combines cross-view ranking constraints with within-view neighborhood structure preservation constraints. Yan *et al.* [46] propose to match images and texts in a joint latent space learned with deep canonical correlation analysis (DCCA) [3]. Nam *et al.* [25] propose dual attention networks (DANs) which attend to specific regions in images and words in text through multiple steps and capture the shared concepts between both modalities, but separate them to provide representations in the embedding space at inference time. Zhang *et al.* [50] propose a cross-modal projection matching (CMPM) loss and a cross-modal projection classification (CMPC) loss to learn cross-modal embeddings.

Pairwise similarity learning paradigm focuses on learning a similarity measure which takes a pair of text description and image as input and outputs their matching score. Traditional methods learn linear similarity measure. Wu *et al.* [43, 44] propose to learn bilinear similarity measure by preserving bi-directional relative semantic similarity. Deep models generate similarity measure by complex variants of CNNs [22] or variants of LSTMs [10, 18], or aggregate similarities of fragments of images and texts [12]. Karpathy *et al.* [12] propose to detect and encode image regions at object level with R-CNN, and then infer the image-text similarity by aggregating the similarity scores of all possible region-word pairs. Ma *et al.* [22] propose to use one image CNN to encode the image content, and one matching CNN to learn the joint representation of image and sentence. Li *et al.* [18] propose a latent co-attention mechanism in which the spatial attention relates each word with corresponding image regions while the latent semantic attention

aligns image-word features. Lee *et al.* [17] propose stacked cross attention (SCAN) to align image regions and text words. It first calculates cosine similarity between all image regions and words of sentence to get attended sentence vectors which are weighted combination of word representations, then it calculates cosine similarity between all attended sentence vectors and image region features and uses LogSumExp pooling or average pooling to obtain the final image-sentence similarity. This paradigm requires the preparation of all image-text pairs for similarity score prediction at the test stage, thus it is computationally demanding and prohibitive for large-scale data.

2.2 Attention Mechanisms

Attention mechanisms, derived from human intuition, allow models to focus on necessary parts of an input where the most relevant information is concentrated. They are usually done by encoding input data based on the importance score assigned to each element. Visual attention models have been successfully applied to various tasks including image classification [42], object detection [42], image generation [49], image captioning [45], etc. The textual attention approaches also benefit sentiment classification [41], neural machine translation [21, 36], sentence summarization [33], etc.

Recently, attention-based models have been proposed for the image-text matching problem. For textual part, sentences are usually divided into words and phrases [39]. For visual part, one line of work selects the feature from the last pooling layer to preserve the spatial information of the original image [10, 25, 39, 45]. Another line detects and encodes image regions at object level with detector [12, 17]. After obtaining fragments of images and texts, attention mechanisms are explored to relate images and texts fragments [12, 17, 25, 39].

3 PROPOSED METHOD

3.1 Problem Formulation

Assume we have a set of images $\mathcal{V} = \{v(1), \cdots, v(N_v)\}$ and a set of texts $\mathcal{T} = \{t(1), \cdots, t(N_t)\}$. The image-text pair information is provided so that we can access the association relations between images and texts. The goal of this paper is to independently embed images and texts into a d-dim common space to facilitate efficient image-text matching and retrieval. In the common space, images and texts can be compared directly using simple distance metric, *e.g.*, the cosine distance. To exploit the fine-grained relation in images and texts, we first represent each image as a set of image features $v(i) = \{x_1^v(i), \cdots, x_{n_v}^v(i)\}, x_j^v(i) \in \mathbb{R}^{d_v}$, and represent each text as a set of token features $t(i) = \{x_1^t(i), \cdots, x_{n_t}^t(i)\}$, $x_j^t(i) \in \mathbb{R}^{d_t}$. Then we exploit the local fragments to obtain global embeddings of images and texts, respectively. The framework of our proposed method is shown in Figure.1. For clarity, we will omit the number of image and text in the rest of this section.

Since both of image and text branches use self-attention mechanism, we first introduce the self attention model. Then we present the embedding network for image and text respectively. Finally, we describe the loss function which we adopt to learn the model.

3.2 Self-Attention

Attention, which is very close to its literal meaning, tells where exactly to attend to. The essence of attention mechanism is that they imitate the human visual perception mechanism. When we humans see a scene, typically we do not scan the entire scene, but always focus on a specific portion according to our needs. In machine learning, an attention function can be described as mapping a query and a set of key-value pairs to an output. The output of attention function is a weighted sum of the value, where the weight matrix, or affinity matrix, is determined by query and its corresponding key.

Since we process a set of image region features, or a set of word features, and expect to embed images and sentences into the joint space without being paired, we can use self-attention which is a special case of the attention mechanism to encode the interaction between fragments of images or texts. In self-attention, queries, keys and values are equal. The self-attention mechanism we apply is similar in spirit to Transformer [36]. The self-attention layer has two sub-layers, *i.e.* multi-head self-attention sub-layer and position-wise feed-forward network sub-layer.

In multi-head self-attention sub-layer, attention is calculated h times, making it to be multi-headed. This is done by respectively projecting the queries (Q), keys (K) and values (V) h times with different learned linear projections. Assume we have a set of fragments $\{f_1, \cdots, f_{n_f}\}, f_i \in \mathbb{R}^{1\times d_f}$, where n_f is the number of fragments, and d_f is the dimensionality of fragments. Packing together theses fragments, we obtain matrix $F = [f_1; \cdots; f_{n_f}] \in \mathbb{R}^{n_f \times d_f}$. The multi-head self-attention sub-layer is computed by:

$$\text{MultiHead}(F) = \text{concat}(\text{head}_1, \ldots, \text{head}_h)W^O, \tag{1}$$

$$\text{head}_i = \text{attention}(FW_i^Q, FW_i^K, FW_i^V), \tag{2}$$

$$\text{attention}(Q, K, V) = \text{softmax}(\frac{QK^\top}{\sqrt{d_k}})V, \tag{3}$$

where parameter matrices $W_i^Q \in \mathbb{R}^{d_f \times d_k}$, $W_i^K \in \mathbb{R}^{d_f \times d_k}$, $W_i^V \in \mathbb{R}^{d_f \times d_{va}}$, $W^O \in \mathbb{R}^{hd_{va} \times d_f}$, $Q \in \mathbb{R}^{n_f \times d_k}$, $K \in \mathbb{R}^{n_f \times d_k}$ and $V \in \mathbb{R}^{n_f \times d_{va}}$. The attention adopted here is the so called "Scaled Dot-Product Attention" that utilizes scaled dot-product to calculate similarity between queries and keys. Note that the parameters are different each time queries, keys and values undergo a linear transformation. Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions.

After the multi-head self-attention sub-layer, to further adjust the fragment representations, the position-wise feed-forward network is applied to each fragment separately and identically:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2, \tag{4}$$

where $x \in \mathbb{R}^{1\times d_x}$, $W_1 \in \mathbb{R}^{d_x \times d_x}$, $W_2 \in \mathbb{R}^{d_x \times d_x}$, $b_1 \in \mathbb{R}^{1\times d_x}$ and $b_2 \in \mathbb{R}^{1\times d_x}$. Similar to the Transformer structure, residual connections followed by layer normalization are also applied around each of the two sub-layers to propagate position information to higher layers.

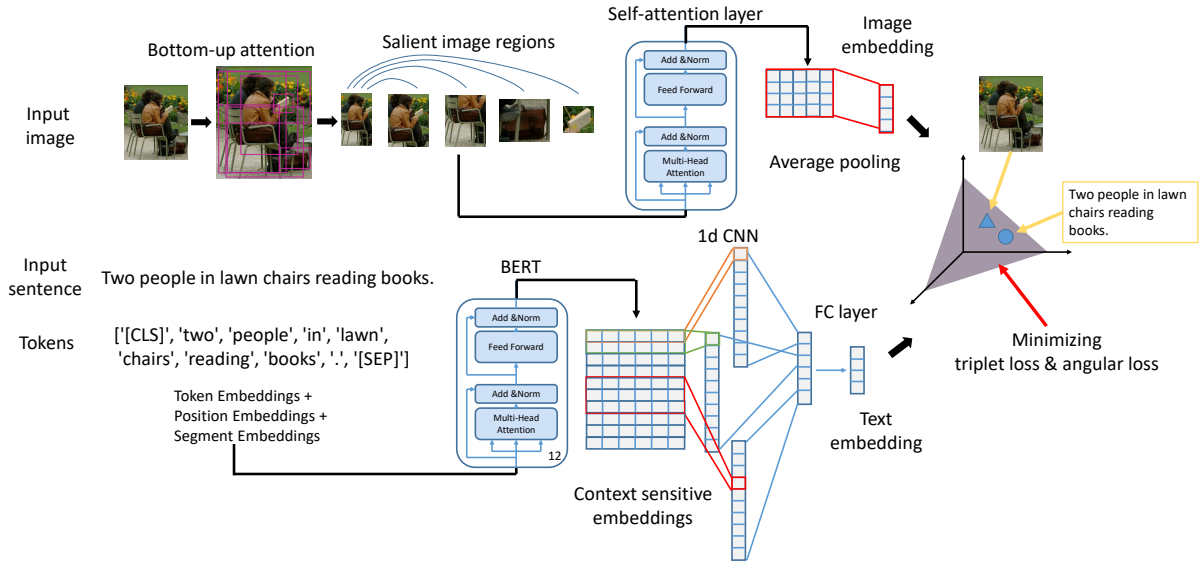

Figure 1: The framework of our method. SAEM consists of two branches, one for image and one for text.

3.3 Image Embeddings

To get visual fragments, one simple solution is selecting the feature from the last pooling layer of CNNs to preserve the spatial information of the original image [10, 25, 39, 45]. However, these approaches just divide image equally in spatial level and do not consider any semantics. Besides, it is very likely to get a lot of fragments containing unimportant background if we divide an image equally, even though these fragments can be filtered out by additional mechanism, they cause unnecessary computation and increase the demands upon algorithm design. Therefore, following [12, 17], we consider to extract objects and other salient image regions using pre-trained detector. Specifically, we employ bottom-up attention [2] analogous to the spontaneous bottom-up attention in human vision system. The bottom-up attention mechanism proposes a set of salient image regions, with each region represented by a pooled convolutional feature vector.

The bottom-up attention mechanism is implemented by Faster R-CNN [32], a two-stage object detection algorithm. In the first stage, region proposals are generated by Region Proposal Network (RPN) which is a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position. In the second stage, the predicted region proposals are further reshaped using an RoI pooling layer which is then used to classify the image within the proposed region and predict the offset values for the bounding boxes.

In this paper, beyond the original Faster R-CNN model which includes classification and bounding box regression outputs for both the RPN and the final object class proposals, we employ the Faster R-CNN model with ResNet-101 [7] pre-trained by Anderson *et al.* [2] on Visual Genomes [16]. Anderson *et al.* [2] add an additional multi-class loss component to train an attribute predictor to provide more semantic information of image patches.

After performing bottom-up attention, an image is represented as a set of image features $v = \{x_1^v, \cdots, x_{n_v}^v\}, x_j^v \in \mathbb{R}^{1\times d_v}$, where n_v is the number of regions, d_v is the dimension of image features, and each image feature encodes a salient region in the corresponding image. We then add a position-wise fully connected layer to transform image features to d-dim vectors $\{y_1^v, \cdots, y_{n_v}^v\}, y_j^v \in \mathbb{R}^{1\times d}$. Packing together image regions, we obtain the matrix $Y^v = [y_1^v; \ldots; y_{n_v}^v] \in \mathbb{R}^{n_v\times d}$.

In order to facilitate similarity calculation at inference time, we need to learn embeddings of images by itself, instead of using complex network which involves text to predict pairwise similarity score. To achieve this, we first exploit an effective mechanism to learn the relation of image regions. Considering that the regions detected by Faster R-CNN in an image do not have a fixed order, previous works [27, 48] usually coarsely organize fragments into a linear sequence and feed into LSTM or biLSTM. Different from these methods, we use the self-attention layer introduced in subsection 3.2 to encode the complex relations of image regions.

With self-attention mechanism, each output fragment can attend to all input fragments, and the distance between each fragment is just one. Thus, our model does not consider any specific order of image regions. After the multi-head self-attention sublayer followed by the layer normalization, we get the output $O^v = [o_1^v; \ldots; o_{n_v}^v] \in \mathbb{R}^{n_v\times d}$:

$$O^v = \text{LayerNorm}(Y^v + (\text{MultiHead}(Y^v))). \tag{5}$$

Then, the position-wise feed-forward network and layer normalization are applied, and the respective output is:

$$z_i^v = \text{LayerNorm}(o_i^v + \text{FFN}(o_i^v)),\ i = 1 \cdots, n_v. \tag{6}$$

After self-attention layer, we get a set of continuous representations $\{z_1^v, \ldots, z_{n_v}^v\}$. Then simple average pooling operation [19] is used to aggregate the representations, resembling bag of visual words model which has shown success in content based image indexing and retrieval [29] in early ages. Thus the image regions are summarized into a compact embedding by average pooling:

$$e^v = \frac{1}{n_v}\sum_i z_i^v,\ i = 1 \cdots, n_v. \tag{7}$$

Finally, L2 normalization is applied to normalize the image embeddings.

3.4 Sentence Embeddings

Following Devlin *et al.* [4], in our model, sentences are tokenized by WordPiece tokenizer, so a sentence t is represented as a sequence of WordPiece tokens $\{x_1^t, \ldots, x_{n_t}^t\}$, where x_k^t is 1-hot encoding representation of the k-th token. We progressively obtain sentence embeddings from low-level tokens.

Motivated by [36] and [4], we adopt Transformer encoder to map an input sequence of tokens $\{x_1^j, \ldots, x_{n_t}^j\}$ to a sequence of continuous representations $\{z_1^j, \ldots, z_{n_t}^j\}$. Specifically, Transformer has an encoder-decoder structure and is based solely on attention mechanism. In our work, we aim to encode sentences to embeddings, thus we only use Transformer encoder that consists of multilayer of self-attention layers introduced in subsection 3.2.

For instantiation, we use the architecture of BERT (Bidirectional Encoder Representations from Transformers) [4], which is designed to pre-train deep bi-directional representations by jointly conditioning on both left and right context in all layers. BERT is pretrained with two unsupervised prediction tasks: the "masked language model" (MLM) and the "next sentence prediction". The masked language model randomly masks some percentage of the input tokens at random, and then predicts only those masked tokens. The next sentence prediction is to predict whether sentence A is the next sentence of B. For the pre-training corpus, BERT uses the concatenation of BooksCorpus and English Wikipedia.

The sequence of continuous representations $\{z_1^j, \ldots, z_{n_t}^j\}$ can also be seen as word embeddings. Unlike Word2Vec [24] and GloVe [28] which are context insensitive, the word embeddings produced by Transformer are context sensitive representations. Context sensitivity means giving different representations according to the sentences. For example, "bank" in the context of rivers or any water body and in the context of finance would not have the same representation.

Since sentence has nature order, after getting the sequence of word representations $Z = [z_1^j; \ldots; z_{n_t}^j]$, we apply 1-dim convolution neural networks [13] to fully exploit the local context information of the sequential features. Specifically, three window sizes, *i.e.*, uni-gram, bi-gram and tri-gram, are used to capture the phrase level information. At each word location, we compute the inner product of the word vectors with filters of three window sizes. The convolutional output using window size s for the k-th word is:

$$p_{s,k} = \text{relu}(W_s z_{k:k+s-1} + b_s),\ s \in \{1, 2, 3\} \tag{8}$$

where W_s is the convolution filter matrix, and b_s is the bias. Before feeding into bi-gram and tri-gram convolutions, the word representations Z are appropriately 0-padded to maintain the length of the sequence after convolution. After obtaining the convolution outputs, we apply max-pooling operation across all word locations:

$$q_s = \max\{p_{s,1}, \cdots, p_{s,n_t}\} \tag{9}$$

We have described the process by which one feature is extracted from one filter. Actually, multiple filters are used in our model. The multiple features are concatenated into a vector and are passed to a fully connected layer followed by L2 normalization to get the final text embedding $e^t \in \mathbb{R}^{1\times d}$.

3.5 Hard-Negative-Based Loss Function

Let e^v denote the embedding of image v, and e^t denote the embedding of text t. We define the scoring function with inner product, *i.e.*, $s(e^v, e^t) = e^v(e^t)^\top$. Since the learned embeddings are scaled to have unit norm, the scoring function is equivalent to cosine similarity.

The network is trained with bi-directional triplet ranking loss, which encourages the matching scores of the truly matched images and sentences to be larger than those of mismatched ones. For a positive pair (v, t), the triplet loss we adopt is:

$$\begin{aligned}\mathcal{L}_{triplet}(v,t) = &\max[0, m - s(e^v, e^t) + s(e^v, e^{\hat{t}})] \\ &+ \max[0, m - s(e^v, e^t) + s(e^{\hat{v}}, e^t)]\end{aligned} \tag{10}$$

where m denotes the margin parameter in triplet loss, $\hat{t}$ denotes a negative sentence for the image v, and $\hat{v}$ a negative image for the text t. Here we consider the hard negatives in a mini-batch, *i.e.*, $\hat{v} = \arg\max_{h\neq v} s(e^h, e^t)$ and $\hat{t} = \arg\max_{h\neq t} s(e^v, e^h)$. Instead of summing over all the negative samples, in practice, using only the hard negatives in a mini-batch leads to better retrieval performance and better computational efficiency.

In addition, we introduce an angular loss [37] which can capture additional local structure of triplet triangles than the triplet loss by using a third-order geometric constraint. Specifically, angular loss encodes the third-order relation inside triplet in terms of constraining the angle at the negative point of triplet triangles. In the original paper of angular loss, N-pair sampling [34] is used to sample a batch containing multiple triplets and a *log-sum-exp* angular loss is optimized for all the samples in the batch. In this paper, we propose to optimize a bi-directional angular loss and apply the hard negative mining on angular loss, leading to the following formulation:

$$\begin{aligned}\mathcal{L}_{angular}(v,t) = &\log[1 + \exp(f(e^v, e^t, e^{\tilde{t}}))] \\ &+ \log[1 + \exp(f(e^t, e^v, e^{\tilde{v}}))]\end{aligned} \tag{11}$$

where $f(a,p,n) = 4\tan^2\alpha(a+p)n^\top - 2(1+\tan^2\alpha)ap^\top$, where a, p, and n indicate embeddings of image or text, and α denotes the angular margin parameter which constrains the angle of the triplet triangle in the angular loss. $\tilde{v} = \arg\max_{h\neq v} f(e^t, e^v, e^h)$ and $\tilde{t} = \arg\max_{h\neq t} f(e^v, e^t, e^h)$ are the hard negatives for angular loss in a mini-batch.

By combining hard-negative-based triplet loss with angular loss, we obtain the overall loss:

$$\mathcal{L}(v,t) = \mathcal{L}_{triplet}(v,t) + \theta\mathcal{L}_{angular}(v,t), \tag{12}$$

where the weight θ controls the importance of angular loss.

In experiment, we observe that angular loss can effectively accelerate the training process. However, it is the scoring function that is used in the final retrieval phase, so triplet loss still plays a more important role than angular loss in practice. Therefore, we propose to change θ as a function of epoch number. We set a large value for θ and then reduce it as the training process goes on, which is similar to the manner of learning rate decay. A further analysis of the loss function can be found in section 4.

3.6 Discussion

As we apply average pooling to get the final image embeddings, our method can be seen as learning n_v individual image embeddings $\{z_1^v, \ldots, z_{n_v}^v\}$ and calculating the similarities between these individual image embeddings and text embedding to get n_v similarities. Then these similarities are aggregated to get the final similarity, which is analogous to the bagging method. In fact, since we use self-attention mechanism to learn individual image embeddings, each individual image embedding takes a region as a key and contains the information of the whole image. Therefore, we call the individual image embeddings as region-centered image embeddings.

4 EXPERIMENTS

4.1 Datasets and Evaluation Metrics

Flickr30K [47] consists of 31,783 images collected from the Flickr website. Each image is accompanied with 5 human annotated sentences. We follow the public splits by [12, 17], using 1,000 images for validation and 1,000 images for testing and the rest for training.

Microsoft COCO [20] consists of 123,287 images, and each image is annotated with five text descriptions. In [12], there are 82,783 training images, 5,000 validation images and 5,000 test images. We follow [17] to add 30,504 images that were originally in the validation set of MSCOCO into the training set. The testing results are reported by averaging over 5 folds of 1,000 test images.

We report the performance of bi-directional cross-modal retrieval tasks: (1) image query versus text database (image-to-text), (2) text query versus image database (text-to-image). We use the commonly used metric Recall@K (K=1, 5, 10), which represents the percentage of the queries where at least one ground-truth is retrieved among the top K results.

4.2 Implementation Details

We implement our architecture in PyTorch framework [26] with an NVIDIA GeForce GTX 1080Ti GPU. We use the Adam [14] optimizer. For the learning rate, we use a small learning rate starting with 0.0001 and decay the learning rate by 0.1 after every 10 epochs. The batch-size is set to 64.

For image branch, we use only 1 self-attention layer which has 16 heads. The image region feature vector extracted by bottom-up attention [2] is 2048-dimensional, so we add a fully-connect layer to transform it to a d-dimensional vector before feeding into the self-attention layer. For the Transformer encoder in text branch, we use the pre-trained weights of BERT model [4] which has 12 self-attention layers, 12 heads, 768 hidden units for each token and 110M parameters in total. For efficient optimization, we fix the weights of Transformer encoder in text branch. In 1-dim convolution neural networks, we use 256 filters for each filter size. For the loss function, we set margin m to 0.2 and angular margin α to 45°. For the combining weight θ, we set it to 0.5 at the beginning, and decay the combining weight by 0.1 after every 5 epochs. Inner product is used to measure similarity on the latent space, which is equivalent to the Euclidean distance since the outputs of the two branches are L2-normalized.

4.3 Performance Comparison

We compare our method with several state-of-the-art methods on Flickr30K and MSCOCO datasets in Tables 1 and 2, respectively.

From Table 1, we can see that our method SAEM achieves the best results on most of the metrics except R@10 on image-to-text task. The promising results confirm the effectiveness of the proposed SAEM. Particularly, when measured by R@1, SAEM outperforms the best baselines by 1.7 and 3.8 on image-to-text task and text-to-image task respectively. The superiority of SAEM can be attributed to its ability to exploit the fine-grained image regions and words in sentence, and to relate fragments using multi-head self attention mechanism.

The performance on the MSCOCO dataset is shown in Table 2. It can be seen from the table that our method is comparable to the best competitor SCAN, and outperforms other compared methods. However, even though SCAN achieves better results on MSCOCO dataset, our SAEM is more efficient and more suitable for the retrieval applications. Considering that the complexity of performing retrieval with N queries and M documents is $O(NM)$, it is important to decrease the time consumed by calculating similarity between data instances. Note that our method embeds image and text into hidden space, and uses inner product to calculate the semantic similarity between image and text, which can facilitate large-scale retrieval at inference time. In comparison, SCAN uses complex cross-attention to derive the similarity of all image-text pairs which needs exhaustive similarity calculation for all image regions and words. Actually, when calculating the similarity between 1,000 testing images and 1,000 testing texts, SCAN takes 467.03s, while our method only takes 0.71s.

Moreover, comparing all methods, we find that methods that introduce prediction of attribute when extracting image features, *i.e.*, SAEM, SCAN and SCO, outperform other methods, indicating the importance of attribute in matching image and text.

4.4 Analysis of Hidden Space Dimension

Furthermore, we conduct experiments on Flickr30K dataset to examine the effect of dimensionality of the hidden space. We show the image-text matching performance with varying dimensions in Table 3. We can see from the table that as the dimension of hidden space increases, the performance of SAEM first increases, then decreases. The best results are obtained by setting the dimension of hidden space to 256 or 512. The experimental results show that larger dimensions does not give better performance, which may be because larger dimensions leads to larger model which is difficult to train. Combining with the common knowledge that lower dimensionality leads to better efficiency, it is better to select an appropriate middle-sized dimensionality of hidden space.

4.5 Analysis of Network Structure

We design some variants to analyze the behavior with different structures. Their front ends are the same, *i.e.*, they all have features of image regions extracted by bottom-up attention and features of tokens extracted by BERT.

- img-fc. We apply linear transformation on features of image regions without self-attention and perform average pooling to get the final embeddings.

Table 1: Bidirectional retrieval results on the Flickr30K dataset compared with state-of-the-art methods.

Method	Image-to-Text			Text-to-Image		
	R@1	R@5	R@10	R@1	R@5	R@10
DVSA (R-CNN, AlexNet) [12]	22.2	48.2	61.4	15.2	37.7	50.5
DCCA (AlexNet, TF-IDF)[46]	27.9	56.9	68.2	26.8	52.9	66.9
DSPE(VGG, Fisher vector) [38]	40.3	68.9	79.9	29.7	60.1	72.1
JGCAR(VGG) [39]	44.9	75.3	82.7	35.2	62.0	72.4
DAN (ResNet) [25]	55.0	81.8	89.0	39.4	69.2	79.1
VSE++ (ResNet) [5]	52.9	87.2	-	39.6	-	79.5
DPC (ResNet) [51]	55.6	81.9	89.5	39.1	69.2	80.9
SCO (ResNet) [11]	55.5	82.0	89.3	41.1	70.5	80.1
CMPM +CMPC (ResNet) [50]	49.6	76.8	86.1	37.3	65.7	75.5
SCAN (Faster R-CNN, ResNet) [17]	67.4	90.3	**95.8**	48.6	77.7	85.2
SAEM-256d	**69.1**	**91.0**	95.1	**52.4**	**81.1**	**88.1**

Table 2: Bidirectional retrieval results on the MSCOCO dataset compared with state-of-the-art methods.

Method	Image-to-Text			Text-to-Image		
	R@1	R@5	R@10	R@1	R@5	R@10
DVSA (R-CNN, AlexNet) [12]	22.2	48.2	61.4	15.2	37.7	50.5
DSPE(VGG, Fisher vector) [38]	50.1	79.7	89.2	39.6	75.2	86.9
JGCAR (VGG) [39]	52.7	82.6	90.5	40.2	74.8	85.7
VSE++ (ResNet) [5]	64.6	-	95.7	52.0	-	92.0
DPC (ResNet) [51]	65.6	89.8	95.5	47.1	79.9	90.0
SCO (ResNet) [11]	69.9	92.9	97.5	56.7	87.5	94.8
CMPM (ResNet) [50]	56.1	86.3	92.9	44.6	78.8	89.0
SCAN (Faster R-CNN, ResNet) [17]	**72.7**	**94.8**	**98.4**	**58.8**	88.4	94.8
SAEM-256d	71.2	94.1	97.7	57.8	**88.6**	**94.9**

Table 3: Effect of different hidden space dimensionality on Flickr30K.

Method	Image-to-Text			Text-to-Image		
	R@1	R@5	R@10	R@1	R@5	R@10
SAEM-64d	62.1	88.0	93.9	47.0	76.3	85.2
SAEM-128d	67.6	89.0	93.8	50.2	78.6	86.8
SAEM-256d	69.1	**91.0**	**95.1**	**52.4**	**81.1**	**88.1**
SAEM-512d	**69.5**	90.6	95.0	52.2	80.1	87.2
SAEM-1024d	66.6	89.4	93.9	51.3	79.0	86.7

- img-cnn. We organize regions into a linear sequence at random, and concatenate them as word embeddings and perform operations similar to 1d-CNN in text branch.
- img-rnn. We organize regions into a linear sequence and sequentially feed all the image regions into biGRU at different time-steps. Then we add the vectors of two directional hidden states at the same time-step as the representation for the corresponding input region. The final embeddings are calculated by averaging all region representations.
- img-satt. This is the same as the image branch in original SAEM.
- txt-fc. We apply linear transformation on features of words, and perform average pooling to get the final embeddings.
- txt-cnn. This is the same as the text branch in original SAEM.
- txt-rnn. Similar to img-rnn, we sequentially feed all tokens of the sentence into biGRU.
- txt-satt. We add a self-attention layer after BERT and use the representation of the beginning token, *, i.e.* <CLS>, as the embeddings of sentence.

The bi-directional retrieval performance of the above variants on Flickr30K dataset is shown in Table 4. When modeling image fragments and words with linear transformation and average pooling, the performance is poor. The performance of SAEM under (img-fc, txt-cnn) is worse than original SAEM (img-satt, txt-cnn), indicating that self-attention mechanism can capture the complex relations of image regions. Moreover, the results of SAEM under (img-cnn, txt-cnn) and SAEM under (img-rnn, txt-cnn) are worse than original SAEM. Note that image regions do not have a natural order, but it needs to organize regions into a linear sequence for CNNs or RNNs, while it is not needed by self-attention. Thus, CNN which exploits local structure is not suitable for image regions, and RNN which relates fragments with different distance performs worse than self-attention. The performance of SAEM under (img-satt, txt-fc) is even worse than SAEM (img-fc, txt-fc), which shows that a good textual representation is important for learning self-attention layer in image branch. The performance of SAEM (img-satt, txt-rnn) is just slightly worse than original SAEM, which shows the power of RNN and 1d-CNN in modeling sequential data.

4.6 Analysis of Loss Function

The loss function in Eq. (12) consists of triplet loss and angular loss. We conduct experiments to see how the two losses affect the matching performance. Also, we show performance without hard negative mining and use 'no hnm' to denote this situation. The experimental results are shown in Table 5.

It can be seen from Table 5 that the performance of only using triplet loss ($\theta = 0$) or angular loss is not good. Specifically, the

Table 4: Effect of different SAEM structures on Flickr30K.

Method	Image-to-Text			Text-to-Image		
	R@1	R@5	R@10	R@1	R@5	R@10
img-fc, txt-fc	54.1	80.4	87.5	39.3	69.8	79.7
img-fc, txt-cnn	64.1	88.3	93.4	45.4	75.3	83.6
img-cnn, txt-cnn	58.3	83.4	91.1	42.9	70.5	79.8
img-rnn, txt-cnn	66.5	88.9	94.0	48.7	77.2	85.4
img-satt, txt-fc	41.0	72.4	81.3	32.5	63.9	75.6
img-satt, txt-satt	59.4	86.1	92.1	46.7	76.8	85.3
img-satt, txt-rnn	69.1	89.8	94.2	51.5	79.5	87.1

Table 5: Effect of loss function on Flickr30K.

Method	Image-to-Text			Text-to-Image		
	R@1	R@5	R@10	R@1	R@5	R@10
only angular	58.2	83.6	89.3	40.9	69.1	76.9
$\theta = 0.1$	68.5	90.9	95.2	51.4	79.2	86.6
$\theta = 0.5$	68.9	89.3	94.8	48.3	76.4	84.1
$\theta = 0$	65.6	88.6	93.8	48.3	76.4	84.1
no hnm	59.4	86.1	92.1	46.7	76.8	85.3

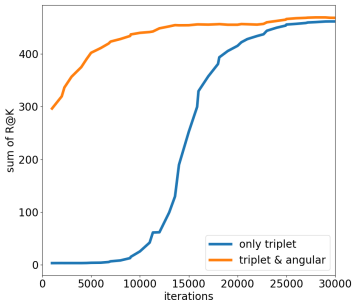

Figure 2: Retrieval performance as a function of iterations.

performance of only using angular loss is worse than only using triplet loss. This may be due to the direct score optimization by minimizing triplet loss that directly encourages scores of matched images and sentences to be larger than those of mismatched ones. From experiment, we observe that angular loss can accelerate the training process. We show the sum of all retrieval metrics as a function of training iterations on two experiments respectively using only triplet loss and using a combination of triplet loss and angular loss, in Figure 2. We can see that with angular loss, our model can achieve a good result very quickly.

4.7 Retrieval Examples

To intuitively show the ranking performance of our method, we illustrate examples of the retrieved texts using image queries and examples of the retrieved images using text queries by SAEM in Figure 3 and Figure 4, respectively. Note that in our settings, an image has five paired sentences, but a sentence only has one paired image. From Figure 3, we can see that almost all of the paired sentences have been retrieved by our method. As can be observed from the examples in Figure 4, besides the ground truth image, other retrieved images also are semantically close to the sentence query. For the first example, we can see that all the retrieved images share the same semantic concept hat with the query, and for the second example, the images all correspond to the same semantics grass.

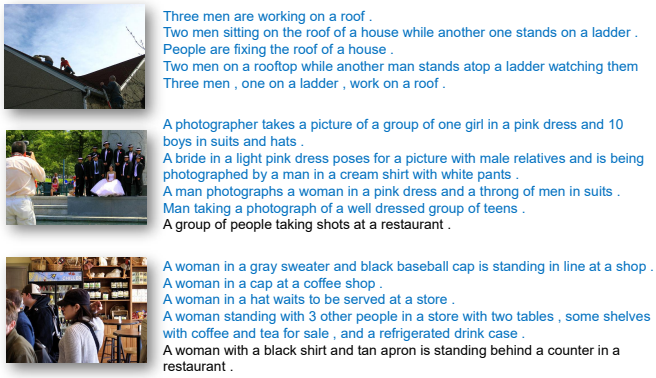

Figure 3: Examples of image queries and the top 5 texts retrieved by the proposed method on Flickr30K dataset. Blue color indicates the paired sentences.

Figure 4: Examples of sentence queries and the top 5 images retrieved by the proposed method on Flickr30K dataset. Blue color indicates the paired images.

Taken together, the results show that our method can ensure that the top ranked results are semantically consistent to the queries.

5 CONCLUSION

In this paper, we propose a novel attention based method for learning image and text embeddings. We use bottom-up attention to extract salient image fragments and apply self-attention mechanism on image fragments to exploit the complex fine-grained visual relation. Then we propose to embed the self-attended visual and textual features into a joint low dimensional space by minimizing the hard-negative-based triplet loss and angular loss. Experimental results on two popular datasets have demonstrated that the proposed method outperforms state-of-the-art approaches with high efficiency. In future work, we will study more advanced technique for extracting and combining image regions and global image information to learn image embeddings.

ACKNOWLEDGMENT

This work was supported in part by National Natural Science Foundation of China: 61672497, 61620106009, U1636214 and 61836002, in part by National Basic Research Program of China (973 Program): 2015CB351800 and in part by Key Research Program of Frontier Sciences of CAS: QYZDJ-SSW-SYS013.

REFERENCES

[1] S. Abhishek, K. Abhishek, H. Daume, and D. W. Jacobs. 2012. Generalized multiview analysis: A discriminative latent space. In *CVPR*. 2160–2167.

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. 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*. 6077–6086.

[3] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. 2013. Deep canonical correlation analysis. In *ICML*. 1247–1255.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[5] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2017. Vse++: Improved visual-semantic embeddings. *arXiv preprint arXiv:1707.05612* 2, 7 (2017), 8.

[6] Yunchao Gong, Qifa Ke, Michael Isard, and Svetlana Lazebnik. 2014. A multi-view embedding space for modeling internet images, tags, and their semantics. *IJCV* 106, 2 (2014), 210–233.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[8] H. Hotelling. 1936. Relations between two sets of variates. *Biometrika* 28, 3/4 (1936), 321–377.

[9] Yan Hua, Shuhui Wang, Siyuan Liu, Anni Cai, and Qingming Huang. 2016. Cross-Modal Correlation Learning by Adaptive Hierarchical Semantic Aggregation. *IEEE Trans. Multimedia* 18, 6 (2016), 1201–1216.

[10] Yan Huang, Wei Wang, and Liang Wang. 2017. Instance-aware image and sentence matching with selective multimodal lstm. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2310–2318.

[11] Yan Huang, Qi Wu, Chunfeng Song, and Liang Wang. 2018. Learning semantic concepts and order for image and sentence matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6163–6171.

[12] Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3128–3137.

[13] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*. 1746–1751.

[14] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations (ICLR)*.

[15] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. 2014. Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models. *CoRR* abs/1411.2539 (2014).

[16] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision* 123, 1 (2017), 32–73.

[17] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. 2018. Stacked cross attention for image-text matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 201–216.

[18] Shuang Li, Tong Xiao, Hongsheng Li, Wei Yang, and Xiaogang Wang. 2017. Identity-aware textual-visual matching with latent co-attention. In *Proceedings of the IEEE International Conference on Computer Vision*. 1890–1899.

[19] Min Lin, Qiang Chen, and Shuicheng Yan. 2013. Network in network. *arXiv preprint arXiv:1312.4400* (2013).

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.

[21] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).

[22] Lin Ma, Zhengdong Lu, Lifeng Shang, and Hang Li. 2015. Multimodal convolutional neural networks for matching image and sentence. In *Proceedings of the IEEE international conference on computer vision*. 2623–2631.

[23] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. 2015. Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN). *ICLR* (2015).

[24] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[25] Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. 2017. Dual attention networks for multimodal reasoning and matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 299–307.

[26] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. (2017).

[27] Yuxin Peng, Jinwei Qi, and Yuxin Yuan. 2018. Modality-specific cross-modal similarity measurement with recurrent attention network. *IEEE Transactions on Image Processing* 27, 11 (2018), 5585–5599.

[28] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[29] Guoping Qiu. 2002. Indexing chromatic and achromatic patterns for content-based colour image retrieval. *Pattern Recognition* 35, 8 (2002), 1675–1686.

[30] V. Ranjan, N. Rasiwasia, and CV Jawahar. 2015. Multi-Label Cross-modal Retrieval. In *ICCV*. 4094–4102.

[31] Nikhil Rasiwasia, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert R. G. Lanckriet, Roger Levy, and Nuno Vasconcelos. 2010. A new approach to cross-modal multimedia retrieval. In *ACM Multimedia*. 251–260.

[32] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems*. 91–99.

[33] Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685* (2015).

[34] Kihyuk Sohn. 2016. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*. 1857–1865.

[35] Guoli Song, Shuhui Wang, Qingming Huang, and Qi Tian. 2017. Multimodal Similarity Gaussian Process Latent Variable Model. *IEEE Trans. Image Processing* 26, 9 (2017), 4168–4181.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.

[37] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. 2017. Deep metric learning with angular loss. In *Proceedings of the IEEE International Conference on Computer Vision*. 2593–2601.

[38] Liwei Wang, Yin Li, and Svetlana Lazebnik. 2016. Learning deep structure-preserving image-text embeddings. In *CVPR*. 5005–5013.

[39] Shuhui Wang, Yangyu Chen, Junbao Zhuo, Qingming Huang, and Qi Tian. 2018. Joint Global and Co-Attentive Representation Learning for Image-Sentence Retrieval. In *2018 ACM Multimedia Conference on Multimedia Conference*. ACM, 1398–1406.

[40] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. 2018. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7794–7803.

[41] Yequan Wang, Minlie Huang, Li Zhao, et al. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*. 606–615.

[42] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. 2018. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 3–19.

[43] Yiling Wu, Shuhui Wang, and Qingming Huang. 2017. Online Asymmetric Similarity Learning for Cross-Modal Retrieval. In *CVPR*. 4269–4278.

[44] Yiling Wu, Shuhui Wang, Guoli Song, and Qingming Huang. 2019. Online Asymmetric Metric Learning With Multi-Layer Similarity Aggregation for Cross-Modal Retrieval. *IEEE Trans. Image Processing* 28, 9 (2019), 4299–4312.

[45] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*. 2048–2057.

[46] Fei Yan and Krystian Mikolajczyk. 2015. Deep correlation for matching images and text. In *CVPR*. 3441–3450.

[47] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics* 2 (2014), 67–78.

[48] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. 2018. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5831–5840.

[49] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. 2018. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318* (2018).

[50] Ying Zhang and Huchuan Lu. 2018. Deep Cross-Modal Projection Learning for Image-Text Matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 686–701.

[51] Zhedong Zheng, Liang Zheng, Michael Garrett, Yi Yang, and Yi-Dong Shen. 2017. Dual-Path Convolutional Image-Text Embedding with Instance Loss. *arXiv preprint arXiv:1711.05535* (2017).