Action-Aware Embedding Enhancement for Image-Text Retrieval

Jiangtong Li, Li Niu*, Liqing Zhang*,

MoE Key Lab of Artificial Intelligence,
Department of Computer Science and Engineering,
Shanghai Jiao Tong University
{keep_moving-lee, ustcnewly}@sjtu.edu.cn, zhang-lq@cs.sjtu.edu.cn

Abstract

Image-text retrieval plays a central role in bridging vision and language, which aims to reduce the semantic discrepancy between images and texts. Most of existing works rely on refined words and objects representation through the data-oriented method to capture the word-object cooccurrence. Such approaches are prone to ignore the asymmetric action relation between images and texts, that is, the text has explicit action representation (i.e., verb phrase) while the image only contains implicit action information. In this paper, we propose Action-aware Memory-Enhanced embedding (AME) method for image-text retrieval, which aims to emphasize the action information when mapping the images and texts into a shared embedding space. Specifically, we integrate action prediction along with an action-aware memory bank to enrich the image and text features with action-similar text features. The effectiveness of our proposed AME method is verified by comprehensive experimental results on two benchmark datasets.

1 Introduction

In recent years, with the prevalence of deep learning and the rapid growth of multimedia data on the internet, vision and language understanding has become more and more important. A large amount of research has been done to bridge the modality gap between vision and language, including image-text retrieval (Faghri et al. 2018; Lee et al. 2018), image captioning (Vinyals et al. 2015), visual question answering (Antol et al. 2015), and visual commonsense reasoning (Zellers et al. 2019). In this paper, we focus on image-text retrieval, which aims to retrieve the texts (*resp.*, images) that describe the most relevant contents for a given image (*resp.*, text) query.

To tackle this problem, a straightforward solution is to directly map images and texts into a shared embedding space, which is regarded as embedding learning paradigm. These approaches learn global representation within each modality and use different techniques like attention mechanism (Nam, Ha, and Kim 2017) or graph convolution networks (Li et al. 2019) to filter out irrelevant information. However, these approaches fail to explore the fine-grained correspondence between image objects and text words, leading to limited performance. Another type of works perform fragment-level matching between objects and words, and aggregate fine-grained similarities as the similarity between images and texts (Chen et al. 2020a; Lee et al. 2018), which is regarded as pair-wise learning paradigm. These methods can capture the fine-grained correspondence between images and texts with state-of-the-art performance on benchmark datasets. Nevertheless, the fine-grained matching between each query and all candidates usually leads to slow retrieval speed, which limits its real-world application.

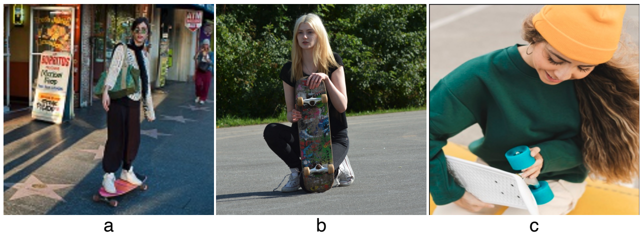


Figure 1: Possible image content for “A woman is <*unk*> her skateboard down the road.” <*unk*> could be “riding”, “holding”, and “checking”, corresponding to (a), (b), and (c), respectively.

Method	Text Retrieval			Image Retrieval		
	R@1	R@5	R@10	R@1	R@5	R@10
VSRN (Li et al. 2019)	68.5	88.4	93.3	50.5	76.6	83.9
w/o verb	68.1	88.9	92.7	49.9	76.8	83.3
w/o noun	11.7	34.6	49.9	9.1	25.3	36.6
IMRAM (Chen et al. 2020a)	71.5	92.7	95.1	52.5	79.0	86.5
w/o verb	70.9	92.4	94.5	52.3	79.1	86.1
w/o noun	12.1	35.1	51.2	9.7	26.3	37.4

Table 1: The experiment results of VSRN (Li et al. 2019) and IMRAM (Chen et al. 2020a) on Flickr30K (Young et al. 2014) with and without verbs/nouns in texts.

Alongside the development of image-text retrieval, the primary concern is to reduce the semantic discrepancy between images and texts. However, most of existing works rely on refined words and objects representation through the data-oriented method to capture the object-word cooccur-

*Corresponding authors.

rence between images and texts, that is, the objects in the image have the corresponding words in the text. Whereas, these methods are prone to ignore the asymmetric action relation between images and texts, that is, texts have explicit action representation (*i.e.* verb) while images only contain implicit action information. In fact, action plays an essential role in image-text retrieval. Given an incomplete sentence "A woman is <*unk*> her skateboard down the road.", where the <*unk*> can be riding, holding, checking and so on. Different verbs correspond to different images, as shown in Figure 1. Nevertheless, in both Flickr30K and Microsoft COCO datasets, "riding" co-occurs with "skateboard" and "man/woman" most frequently, which makes the retrieved images biased towards Figure 1 (a) once "man/woman" and "skateboard" appear in the query text.

To verify our hypothesis, we further experiment on Flickr30K (Young et al. 2014) with the approaches from both embedding learning paradigm (*i.e.*, VSRN (Li et al. 2019)) and pair-wise learning paradigm (*i.e.*, IMRAM (Chen et al. 2020a)) in Table 1. To explore the effect of verbs (*resp.*, nouns) towards these approaches, all the verbs (*resp.*, nouns) detected by Stanford CoreNLP model (Manning et al. 2014) are replaced by "<*unk*>" in "w/o verb" (*resp.*, "w/o noun") setting. As shown in Table 1, we can find that the explicit action information in texts has little effect on image-text retrieval (the performance gap is between ± 0.7), whereas the object information in texts determine the retrieval by a large margin. The experiment results further indicate that these approaches mainly rely on the object-word cooccurrence and tend to ignore the asymmetric action relation between images and texts. Therefore, we propose to emphasize the action information to learn action-aware embedding for image-text retrieval. In fact, GSNM (Liu et al. 2020) and CVSE (Wang et al. 2020a) have already tried to combine the object and action as a whole for image-text matching, however, the experiments in Table 1 reveal that the action information is the part that is in desperate need to be emphasized. Therefore, we choose to enhance the action representation on purpose.

In this paper, we propose Action-aware Memory-Enhanced embedding (AME) method for image-text retrieval, as illustrated in Figure 2. In detail, we first build an action-aware memory bank, which utilizes the explicit action information (*i.e.*, verb phrase) as the key and their corresponding texts as values. Afterwards, we employ a shared transformer encoder (Vaswani et al. 2017) to extract fragment-level image (*resp.*, text) features and image (*resp.*, text) embeddings. Then an action predictor is applied to image (*resp.*, text) embeddings to obtain the image (*resp.*, text) action tags, which are defined as verb phrases. Next, the obtained action tags are used to search for action-similar texts from the action-aware memory bank, which are integrated with fragment-level image and text features to produce the action-aware embeddings. The action-awareness of our proposed method is reflected in two aspects: 1) we use the image and text embeddings to predict their corresponding action tags; 2) we incorporate action-similar texts from action-aware memory bank with fragment-level image and text features to learn enhanced action-aware embeddings.

The effectiveness of our AME method is corroborated by comprehensive experiments on two benchmark datasets. Our main contributions are summarized as follows:

- To emphasize the contribution of action information, we propose to learn action-aware embedding for image-text retrieval task.
- We propose a novel AME approach, which incorporates action-similar texts from the memory bank for action-aware embedding enhancement.
- Comprehensive experiments on two large-scale benchmark datasets reveal that our method significantly outperforms the state-of-the-art methods.

2 Related Work

2.1 Image-Text Retrieval

The key issue of image-text retrieval is to reduce the semantic discrepancy between images and texts. For this purpose, existing works can be categorized into two groups, embedding learning paradigm (Frome et al. 2013; Faghri et al. 2018; Kiros, Salakhutdinov, and Zemel 2014; Wang, Li, and Lazebnik 2016; Li et al. 2019; Zhang and Lu 2018; Gu et al. 2018; Vendrov et al. 2016; Nam, Ha, and Kim 2017; Wu et al. 2019; Weikuo et al. 2019) and pair-wise learning paradigm (Karpathy, Joulin, and Fei-Fei 2014; Ma et al. 2015; Huang, Wang, and Wang 2017; Wu, Wang, and Huang 2017; Niu et al. 2017; Wang et al. 2020b; Lee et al. 2018; Chen et al. 2020a; Chen and Luo 2020; Wang et al. 2018; Li et al. 2017; Wang et al. 2019a,b,c; Chen et al. 2019; Liu et al. 2019; Wehrmann, Kolling, and Barros 2020; Qu et al. 2020; Wei et al. 2020; Qu et al. 2021; Li et al. 2021).

Embedding Learning Paradigm: The embedding learning methods aim to learn a modal-invariant and representative embedding for each image and text. Faghri et al. (2018) paid attention to the hardest negative with the triplet ranking loss. Wu et al. (2019) applied self-attention layers to discover the relation among regions (*resp.*, words) in images (*resp.*, texts) more accurately. Besides, Li et al. (2019) performed reasoning with Graph Convolutional Networks to generate features with visual-semantic relation. To further exploit the relation within images and texts, Qu et al. (2020) built a gated self-attention and a multi-view summarization modules for intra-modal representation and cross-modal matching.

Pair-wise Learning Paradigm: The pair-wise learning methods aim to calculate the similarity between each image-text pair more accurately with fine-grained alignment. To capture structure information in images and texts, Liu et al. (2020) utilized extra information to parse images and texts into graphs, and adopted the Graph Structured Network to match image graphs with text graphs. With the help of the cross attention networks, Lee et al. (2018) obtained the image (*resp.*, text) features by attending each region (*resp.*, word) feature to all word (*resp.*, region) features, while Chen et al. (2020a) proposed IMRAM to match fragments across different modalities iteratively. To combine the intra-modal reasoning and cross-modal alignment modules together, Qu et al. (2021) proposed a dynamic router with the capability to choose the path of different modality interactions.

Some works have also attempted to incorporate action information within the modeling process. (Liu et al. 2020) correlated the object, action, and attribute in a same graph and matched them with GCN. However, the image region representations from Faster R-CNN still cannot model the action information in image explicitly. Wang et al. (2020a) exploited consensus-aware information in extra related datasets and enabled the embedding to predict consensus-level concepts, including object, action and property, which only roughly predict the semantic concept. Unlike the above methods, our AME not only enables action tag prediction, but also utilizes the action-similar texts to enhance the action-aware embedding.

2.2 Memory-Enhanced Network

Memory-enhanced network was proposed by (Weston, Chopra, and Bordes 2014), which targets on enhancing the long-term memory. (Miller et al. 2016) developed a key-value memory network to utilize prior knowledge. After that, memory-enhanced network has become popular in the fields of computer vision (Chen et al. 2020b) and natural language processing (Wang et al. 2016).

Memory-enhanced network has also been widely used in multi-modal modeling. To name a few, Huang and Wang (2019) aligned the cross-modal information in memory bank for few-shot image-text retrieval. Song, Wang, and Tan (2018) proposed a category-based memory bank for cross-modal retrieval. Ji et al. (2020) stored inter- and intra-modal information in memory bank to narrow the gap between two modality. Note that the memory banks used in these works either only captured category information or only utilized local information. In contrast, our work utilizes global memory to facilitate action-aware embedding enhancement.

3 Methodology

In this section, we elaborate on Action-aware Memory-Enhanced embedding (AME) method for image-text retrieval, as illustrated in Figure 2. In Sec. 3.1, we will present the problem definition and notation. In Sec. 3.2, we will detail our AME method, revealing how to emphasize the action information for action-aware embedding. In Sec. 3.3, we will describe the loss functions.

3.1 Problem Definition

Suppose we have a set of training images $\{\mathbf{x}^i_1, ..., \mathbf{x}^i_{N_i}\}$ and a set of training texts $\{\mathbf{x}^t_1, ..., \mathbf{x}^t_{N_t}\}$ with provided matching correspondence (each image has several matched texts), where N_i and N_t are the number of images and texts, respectively. The goal of image-text retrieval is to learn a representation model, which encourages the matched image-text pairs closer than mismatched image-text pairs in the shared embedding space. For clarity, in the remainder of this paper, we will omit the index number of images/texts, and all similarity is measured under cosine similarity. We use $\mathbf{1}$ to denote an all-one column vector.

The overall structure of our proposed AME method is illustrated in Figure 2, which includes three main components: (1) fragment-level representation learning; (2) action-aware memory search; (3) action-aware representation learning. During the first phase, we represent images and texts by corresponding fragment-level features and embeddings through a transformer encoder. Based on the embedding of each image and text, we employ an action predictor to get action tags, with which we further search for some action-similar texts to obtain the action-similar text features. Finally, with the assistance of another transformer encoder, we fuse the action-similar text features with the fragment-level image (*resp.* text) features to obtain the enhanced action-aware image (*resp.* text) embedding.

3.2 Action-Aware Memory-Enhanced Embedding

Fragment-level Representation Learning. Given an image $\mathbf{x}^i$ and a text $\mathbf{x}^t$ as a pair of inputs, we first represent each of them at fragment level, *i.e.*, representing them as a sequence of feature vectors. For each image, to capture the fragment-level regional information, we follow (Lee et al. 2018) to use Faster R-CNN model (Ren et al. 2015) to extract the region features, which employs bottom-up attention (Anderson et al. 2018) to provide convolutional feature for each image region. Therefore, each image is represented as a sequence of image region features $\mathbf{I} \in \mathbb{R}^{n_i \times d_i} = [\mathbf{i}_1, ..., \mathbf{i}_{n_i}]$ ordered by the confidence score, where n_i is the number of regions and d_i is the region feature dimension. For each text, we design two types of settings, 1) we apply word embedding (Pennington, Socher, and Manning 2014) along with Bi-GRU (Chung et al. 2014) to extra word features of each text; 2) we employ pre-trained BERT (Devlin et al. 2019) to generate token features of each text, through any of which we can represent each text as a sequence of features $\mathbf{T} \in \mathbb{R}^{n_t \times d_t} = [\mathbf{t}_1, ..., \mathbf{t}_{n_t}]$, where n_t is the number of words/tokens and d_t is feature dimension.

To encourage the information prorogation among regions (*resp.*, words) within each sequence of image (*resp.*, text) features, we apply a transformer encoder, which encourages the intra-modal information propagation, to learn better image (*resp.*, text) representation.

First, we project region features $\mathbf{I}$ (*resp.* word features $\mathbf{T}$) into the same dimension d, which is denoted as $\mathbf{I}^d$ (*resp.* $\mathbf{T}^d$). Then, we employ a transformer encoder E^e, where the input query, key, and value are all $\mathbf{I}^d$ (*resp.*, $\mathbf{T}^d$), and obtain fragment-level image (*resp.*, text) features $\mathbf{I}^e$ (*resp.*, $\mathbf{T}^e$):

$$\mathbf{I}^e = E^e(\mathbf{I}^d, \mathbf{I}^d, \mathbf{I}^d), \quad \mathbf{T}^e = E^e(\mathbf{T}^d, \mathbf{T}^d, \mathbf{T}^d). \qquad (1)$$

E^e utilizes the feature-level similarity to encourage the intra-modal information propagation among region (*resp.*, word) features within each image (*resp.*, text) to learn better fragment-level image (*resp.*, text) features $\mathbf{I}^e$ (*resp.*, $\mathbf{T}^e$). We utilize a siamese transformer encoder, where images and texts share the same transformer encoder.

2. Action-aware Memory Search

1) Action Prediction. In texts, the action information is usually reflected by verbs. However, directly extracting the verb as the action tag may be ambiguous. For example, “take” has different meanings in “I will take you to the room” and “I can take the basketball this afternoon”. Therefore, we choose the verb phrase as the action tag to represent the action information in each text. To predict the action tags

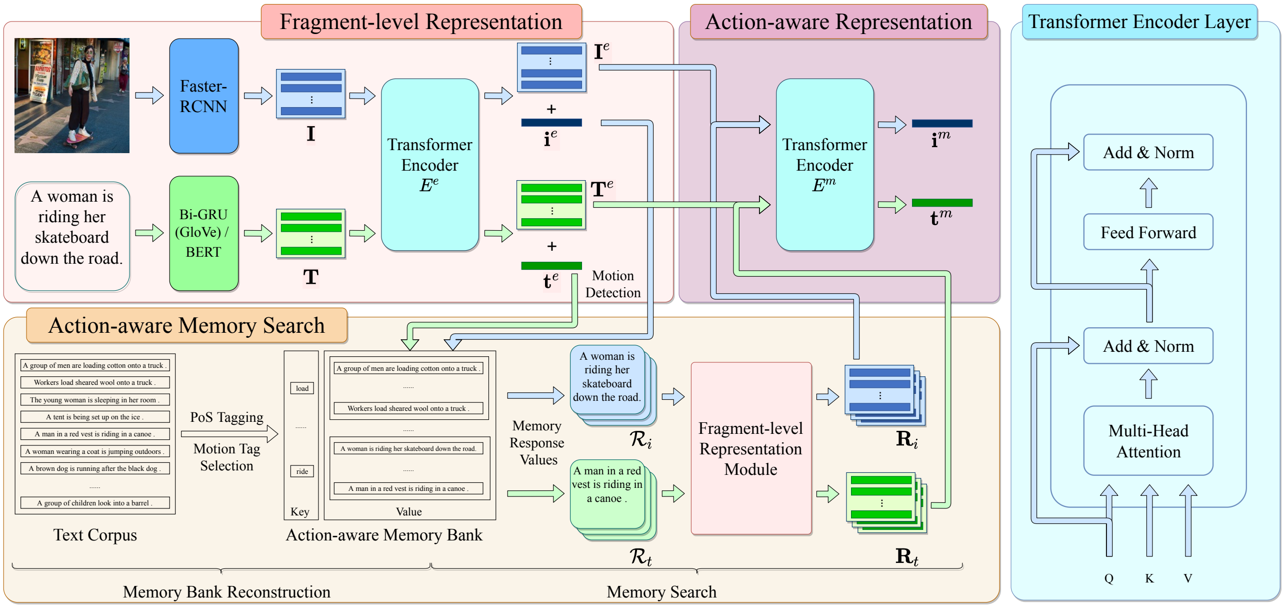

Fragment-level Representation
Faster-RCNN
$\mathbf{I}$
Transformer Encoder E^e
$\mathbf{I}^e$
+
$\mathbf{i}^e$
A woman is riding her skateboard down the road.
Bi-GRU (GloVe) / BERT
$\mathbf{T}$
$\mathbf{T}^e$
+
$\mathbf{t}^e$
Motion Detection
Action-aware Representation
Transformer Encoder E^m
$\mathbf{i}^m$
$\mathbf{t}^m$
Transformer Encoder Layer
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Q
K
V
Action-aware Memory Search
A group of men are loading cotton onto a truck .
Workers load sheared wool onto a truck .
The young woman is sleeping in her room .
A tent is being set up on the ice .
A man in a red vest is riding in a canoe .
A woman wearing a coat is jumping outdoors .
A brown dog is running after the black dog .
......
A group of children look into a barrel .
Text Corpus
PoS Tagging
Motion Tag Selection
load
......
ride
Key
A group of men are loading cotton onto a truck .
......
Workers load sheared wool onto a truck .
......
A woman is riding her skateboard down the road.
......
A man in a red vest is riding in a canoe .
Value
Action-aware Memory Bank
Memory Response Values
A woman is riding her skateboard down the road.
$\mathcal{R}_i$
A man in a red vest is riding in a canoe .
$\mathcal{R}_t$
Fragment-level Representation Module
$\mathbf{R}_i$
$\mathbf{R}_t$
Memory Bank Reconstruction
Memory Search

Figure 2: The flowchart of our AME method. We first adopt Bi-GRU(GloVe) (Chung et al. 2014) or BERT (Devlin et al. 2019) (*resp.* Faster R-CNN (Ren et al. 2015) (bottom-up attention (Anderson et al. 2018))) to extract word (*resp.*, region) features from texts (*resp.*, images), which are fed into transformer encoder to obtain text (*resp.*, image) embeddings. Then, action-aware memory search and action-aware representation are performed to enrich text (*resp.*, image) features with action information, resulting in action-aware text (*resp.*, image) embeddings. The structure of transformer encoder layer is shown on the right.

of each image and text, we first correlate the action tags with each image and text in the training set. Specifically, given a text containing several phrase chunks, we extract the verb phrase based on the definition of dependency parsing, using the off-the-shelf Stanford CoreNLP (Manning et al. 2014), where the resultant verb phrases are regarded as the action tags for each text. For example, in sentence "A man who stands in the playground, is holding a racket.", the following will be selected as action tags: "stands", "who stands", "stands in", "stands in the playground", "is", "holding", "is holding", "man is", "man is holding", "A man is", "A man is holding", "is holding a racket", "holding a racket". To reduce redundancy, we replace all the words in verb phrase with their stems, remove all the articles (a/an/the), adjective, and adverb, and delete the verb phrases appearing less than 5 times in the whole dataset. After that, in the aforementioned example, only "stand", "stand in", "stand in playground", "be", "hold", "be hold" will be kept. For images, it is hard to extract the action tags directly, and hence we exploit matched image-text pairs in the training set to extract the image action tags. In detail, for each image, the action tags are generated by collecting all the action tags belonging to its matched texts.

In the testing stage, the action tags do not exist in test images and directly extracting verb phrase from test texts may ignore some synonyms. Thus, in the training stage, we train an action predictor based on fragment-level image and text features. In detail, we first adopt a max-pooling layer to aggregate fragment-level image $\mathbf{I}^e$ (*resp.*, text $\mathbf{T}^e$) features (choosing the max value in each dimension) as the image $\mathbf{i}^e$ (*resp.*, text $\mathbf{t}^e$) action embedding. Then we employ action predictor to predict the action tags, which is formulated as:

$$\mathbf{m}^{i,e} = \sigma(\mathbf{W}^e\mathbf{i}^e + \mathbf{b}^e), \quad \mathbf{m}^{t,e} = \sigma(\mathbf{W}^e\mathbf{t}^e + \mathbf{b}^e), \qquad (2)$$

where $\mathbf{W}^e$ and $\mathbf{b}^e$ are the parameters of the action predictor for both images and texts, σ is the sigmoid activation function. Here, we treat the prediction task as a multi-label classification task with binary cross-entropy loss, which will be elaborated in Sec. 3.3.

2) Memory Search. To emphasize the action information, we attempt to enhance the image and text features with action-similar texts. For this purpose, we construct an action-aware memory bank to store all training texts. To facilitate the memory search, we collect action tags from all training texts in the same way as for action prediction. We treat each action tag as a key and all the texts containing this action tag as values, which are stored in the action-aware memory bank. Note that here we only describe the memory search procedure for images, as it goes the same for texts. After using action predictor to get the action scores $\mathbf{m}^{i,e}$ of an image, we filter out the action tags with scores below the threshold τ_m, and then select the action tags with top-n_m scores from the remaining action tags as the image action tag candidates $\mathcal{M}^i$. If the number of remaining action tags is smaller than n_m, all of them are maintained as the image action tag candidates $\mathcal{M}^i$.

To get the action-similar texts from the action-aware memory bank, we search the memory bank with the predicted action tags. For each action tag in the action tag candidate $\mathcal{M}^i$, we get an individual action-similar text set by using each action tag as query to search from the memory bank. Then, the union of all these action-similar text sets is regarded as the action-similar texts of the action tag candidates $\mathcal{M}^i$. From all the action-similar texts, we randomly select n_r texts as a set of memory response texts $\mathcal{R}^i$. To facilitate action-aware embedding enhancement, we use the fragment-level features of action-similar text in $\mathcal{R}^i$ as a set of memory response features $\mathbf{R}^i$.

Action-aware Representation Learning. Now for an input image $\mathbf{x}^i$, we have its fragment-level features $\mathbf{I}^e$ and action-aware memory response features $\mathbf{R}^i = \{\mathbf{R}^i_1, \dots, \mathbf{R}^i_{n_r}\}$. To obtain the action-aware representation, we fuse each item of the memory response features $\mathbf{R}^i_j$ with the fragment-level features $\mathbf{I}^e$ through another transformer encoder E^m, and then calculate the average of outputs as the action-aware image features $\mathbf{I}^m$:

$$\mathbf{I}^m_j = E^m(\mathbf{I}^e, \mathbf{R}^i_j, \mathbf{R}^i_j), \quad \mathbf{I}^m = \frac{1}{n_r}\sum_{j=1}^{n_r} \mathbf{I}^m_j. \qquad (3)$$

Finally, the same max-pooling layer is employed to aggregate the action-aware image features $\mathbf{I}^m$ into an action-aware image embedding $\mathbf{i}^m$.

Note that the memory response features only concern the action information predicted by the action predictor, which may contain some irrelevant object descriptions. For example, given the verb phrase "chase on field", the memory response texts could be either "A lion is chasing a zebra on the field." or "Three boys are chasing each other on the field.". Therefore, the memory response features could bring some noisy information into the final representation. However, the function of transformer encoder E_m is to exploit the similarity between each item of the memory response features $\mathbf{R}^i_j$ and image features $\mathbf{I}^e$ to aggregate relevant parts and filter out irrelevant parts in $\mathbf{I}^e$, which prevent the noise in memory response features from being directly introduced to the action representation.

Analogous to input image, we can also use an input text $\mathbf{x}^t$ to obtain corresponding memory response features $\mathbf{R}^t$ and the enhanced action-aware text embedding $\mathbf{t}^m$ via the same transformer encoder E^m.

3.3 Loss and Retrieval

Loss Function To enforce the distance of matched image-text pairs to be closer than mismatched ones, we use triplet ranking loss (Li et al. 2019) in both fragment-level embedding space and action-aware embedding space. Following (Faghri et al. 2018), we employ the hardest negatives, *i.e.*, the negatives closest to each training query. For a positive pair ($\mathbf{t}$,$\mathbf{i}$), we can find the hardest negative image $\hat{\mathbf{i}}$ and the hardest negative text $\hat{\mathbf{t}}$. Then, the triplet loss is defined as

$$\begin{aligned}\mathcal{L}_{tri}(\mathbf{i}, \mathbf{t}) =& [\beta - S(\mathbf{i}, \mathbf{t}) + S(\mathbf{i}, \hat{\mathbf{t}})]_+ \\ &+ [\beta - S(\mathbf{i}, \mathbf{t}) + S(\hat{\mathbf{i}}, \mathbf{t})]_+,\end{aligned} \qquad (4)$$

where β serves as a margin parameter and $[x]_+ = \max(x, 0)$. $S(\cdot,\cdot)$ stands for cosine similarity. For computational efficiency, rather than selecting the hardest negatives in entire training set, we use the hardest one in each mini-batch.

Considering that each image and text usually contains multiple action tags, we treat action prediction for each action tag as an independent binary classification task:

$$\begin{aligned}\mathcal{L}_{bce}(\mathbf{m}, \bar{\mathbf{m}}) = -\sum_{k=1}^{n_{mt}} (\bar{m}_k \log(m_k) + \\ (1 - \bar{m}_k)\log(1 - m_k)),\end{aligned} \qquad (5)$$

where n_{mt} is the size of all action tags, $\bar{\mathbf{m}}$ is the ground-truth binary label vector for n_{mt} action tags, $\bar{m}_k$ (*resp.*, m_k) is the k-th entry of $\bar{\mathbf{m}}$ (*resp.*, $\mathbf{m}$). Note that we omit the supercripts for $\bar{\mathbf{m}}_k$ and $\mathbf{m}_k$ here.

In summary, the final training objective is defined as:

$$\begin{aligned}\mathcal{L} =& \lambda_1 \mathcal{L}_{tri}(\mathbf{i}^m, \mathbf{t}^m) \\ &+ \lambda_2(\mathcal{L}_{bce}(\mathbf{m}^{i,e}, \bar{\mathbf{m}}^i) + \mathcal{L}_{bce}(\mathbf{m}^{t,e}, \bar{\mathbf{m}}^t)),\end{aligned} \qquad (6)$$

where λ_1 and λ_2 aim to balance different loss terms.

Retrieval During retrieval, we concatenate the action-aware embeddings with the action scores together as the final embedding:

$$\mathbf{i}^c = [\mathbf{i}^m, \mathbf{m}^{i,e}], \qquad \mathbf{t}^c = [\mathbf{t}^m, \mathbf{m}^{t,e}], \qquad (7)$$

where $[\cdot,\cdot]$ represents the concatenation of two vectors.

4 Experiment

4.1 Dataset and Evaluation Metrics

Dataset We evaluate our AME method and all the other baselines on two large-scale benchmark datasets: Flickr30K (Young et al. 2014) and Microsoft COCO (Lin et al. 2014).

Flickr30K (Young et al. 2014) consists of 31,000 images from the Flickr website. Each image is associated with five human annotated texts. We follow the split in (Lee et al. 2018), by using 1,000 images for validation, 1,000 images for testing, and 29,000 images for training.

Microsoft COCO (Lin et al. 2014) originally consists of 82,783 training images and 40,504 validation images, and each image is annotated with five texts. Following the split in (Lee et al. 2018), we select 5,000 validation images and 5,000 test images from the original validation set and then add the rest 30,504 images into training set. The test results are reported for averaging over five folds of 1K test images (Li et al. 2019). The test results for 5K test images can be found in Supplementary.

Evaluation Metrics Specifically, we adopt Recall at K (R@K) to measure the performance of the bi-directional retrieval, *i.e.*, retrieving texts given an image (Text Retrieval) and retrieving images given a text (Image Retrieval). We report R@1, R@5, and R@10 on both datasets. We also report the "mR" criterion, the average of all six recall rates of R@K, which provides a more comprehensive evaluation to testify the performance. More details about implementation can be found in Supplementary.

4.2 Comparison with Existing Methods

To justify the effectiveness of our proposed method, we compared it with fourteen prior methods on Flickr30, Microsoft COCO (1K), and Microsoft COCO (5K), which include five embedding learning models and nine pair-wise

Learning Paradigm	Method	Flickr30							Microsoft COCO (1K)						
		Text Retrieval			Image Retrieval			mR	Text Retrieval			Image Retrieval			mR
		R@1	R@5	R@10	R@1	R@5	R@10		R@1	R@5	R@10	R@1	R@5	R@10	
Pair-wise Learning	CAMP	68.1	89.7	95.2	51.5	77.1	85.3	77.8	72.3	94.8	98.3	58.5	87.9	95.0	84.4
	PFAN*	70.0	91.8	95.0	50.4	78.7	86.1	78.7	76.5	96.3	**99.0**	61.6	89.6	95.2	86.4
	DP-RNN	70.2	91.6	95.8	55.5	81.3	88.2	80.5	75.3	95.8	98.6	62.5	89.7	95.1	86.2
	IMRAM	74.1	93.0	96.6	53.9	79.4	87.2	80.7	76.7	95.6	98.5	61.7	89.1	95.0	86.1
	GSNM*	76.4	94.3	97.3	57.4	82.3	89.0	82.8	78.4	96.4	98.6	63.3	90.1	95.7	87.1
	ADAPT*	76.6	95.4	97.6	60.7	86.6	92.0	84.8	76.5	95.6	98.9	62.2	90.5	96.0	86.6
	DIME*†	81.0	**95.9**	98.4	63.6	88.1	93.0	86.7	78.8	96.3	98.7	64.8	**91.5**	**96.5**	87.8
Embedding Learning	SAEM†	69.1	91.0	95.1	52.4	81.1	88.1	79.5	71.2	94.1	97.7	57.8	88.6	94.9	84.0
	VSRN*	71.3	90.6	96.0	54.7	81.8	88.2	80.4	76.2	94.8	98.2	62.8	89.7	95.1	86.1
	CVSE	73.5	92.1	95.8	52.9	80.4	87.8	80.4	74.8	95.1	98.2	59.9	89.4	95.2	85.5
	CAMERA*†	78.0	95.1	97.9	60.3	85.9	91.7	84.8	77.5	96.3	98.8	63.4	90.9	95.8	87.1
	AME	74.9	93.5	97.0	58.9	84.7	90.2	83.2	77.1	95.4	98.3	62.8	89.4	95.1	86.4
	AME*	77.1	95.1	97.3	61.2	86.1	91.5	84.7	78.5	96.1	98.7	63.7	90.1	95.6	87.1
	AME†	78.4	95.4	97.8	62.1	86.8	91.9	85.4	78.6	96.0	98.6	64.2	90.3	95.7	87.2
	AME*†	**81.9**	**95.9**	**98.5**	**64.6**	**88.7**	**93.2**	**87.1**	**79.4**	**96.7**	98.9	**65.4**	91.2	96.1	**87.9**

Table 2: Comparison with existing models on Flickr30K and Microsoft COCO (1K). The symbol '$*$' refers to the ensemble result and the symbol '†' refers to the pre-trained BERT text embedding. The state-of-the-art results are highlighted in bold.

Learning Paradigm	Method	Microsoft COCO (5K)						
		Text Retrieval			Image Retrieval			mR
		R@1	R@5	R@10	R@1	R@5	R@10	
Pair-wise Learning	CAMP	50.1	82.1	89.7	39.0	68.9	80.2	68.3
	IMRAM	53.7	83.2	91.0	39.6	69.1	79.8	69.4
	DIME*†	59.3	**85.4**	91.9	43.1	**73.0**	**83.1**	72.6
Embedding Learning	VSRN*	53.0	81.1	89.4	40.5	70.6	81.1	69.3
	CAMERA*†	55.1	82.9	91.2	40.5	71.7	82.5	70.6
	AME	54.0	82.1	90.7	40.1	70.2	80.5	69.6
	AME*	56.4	83.8	91.7	41.5	71.1	81.4	71.0
	AME†	57.1	83.5	91.6	42.2	71.7	82.0	71.3
	AME*†	**59.9**	85.2	**92.3**	**43.6**	72.6	82.7	**72.7**

Table 3: Comparison with existing models on Microsoft COCO (5K). The symbol '$*$' refers to the ensemble result and the symbol '†' refers to the pre-trained BERT text embedding. The state-of-the-art results are highlighted in bold.

learning models. For embedding learning models, we compare with VSRN (Li et al. 2019), SAEM (Wu et al. 2019), CVSE (Wang et al. 2020a), and CAMERA (Qu et al. 2020), where CVSE (Wang et al. 2020a) adopted a huge knowledge base for cross-modal representation. For pair-wise learning models, we compare with CAMP (Wang et al. 2019c), PFAN (Wang et al. 2019b), DP-RNN (Chen and Luo 2020), IMRAM (Chen et al. 2020a), GSNM (Liu et al. 2020), ADAPT (Wehrmann, Kolling, and Barros 2020), and DIME (Qu et al. 2021), where GSNM adopted extra semantic information to build sparse text graph. All the methods applied Faster-RCNN (Ren et al. 2015) to extract image region features. For fair comparison, we further divide the experiment setting into four different conditions based on the initialization of text embedding and whether using ensemble models.

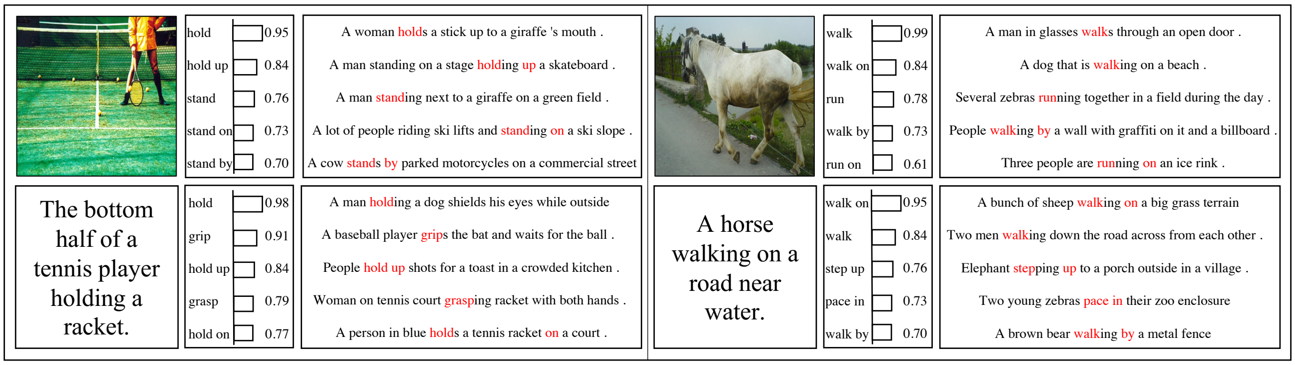
The performance on Flickr30K and Microsoft COCO (1K) are shown in Table 2 and the performance on Microsoft COCO (5K) are shown in Table 3, where our AME method outperforms all the existing methods on corresponding conditions in terms of text retrieval (R@1) and image retrieval (R@1). Compared with the embedding learning methods, our proposed AME method outperforms them by a large margin. Specifically, the performance gain of our method (AME*†) over CAMERA*† is 3.9%, 1.9% and 4.7% (*resp.*, 4.3%, 2.0% and 3.0%) on Flickr30K, Microsoft COCO (1K) and Microsoft COCO (5K) in terms of text retrieval (*resp.*, image retrieval) R@1. Compared with the pair-wise learning methods, our proposed AME method can still gain improvement in all R@1, which indicates that the proposed memory-bank can enhance the embedding with action-aware cross-modal information. Note that the improvement of our AME*† over DIME*† is not quite large, however, the pair-wise learning paradigm make the inference speed of the DIME much slower than our proposed AME method.

Regarding the performance on different datasets, we can find that on Flickr30K, the gain from the model ensemble and pre-trained BERT text embedding is more significant than that in Microsoft COCO (1K). We conjecture that images in Microsoft COCO have much large training set which compromises the enhancement from both model ensemble and pre-trained BERT text embedding. Moreover, by comparing the improvement of our method on both datasets, we can also find that the improvement on Flickr30K is more significant. We suspect that images in Microsoft COCO have fewer objects and simpler relation, which compromises the enhancement from fine-grained alignment and fusion.

4.3 Ablation Study

By taking Flickr30K as an example, we analyze the impact of memory bank, action score, action tag candidate size n_m, and memory response size n_r. Text retrieval R@1, image retrieval R@1, and "mR" under single model and random initialized text embedding are reported in this section.

Effect of Memory Banks and Action score As shown in Table 4, we can find that the memory bank is essential to the performance (row 1 v.s. row 3), without which the per-



hold 0.95
hold up 0.84
stand 0.76
stand on 0.73
stand by 0.70

A woman holds a stick up to a giraffe 's mouth .
A man standing on a stage holding up a skateboard .
A man standing next to a giraffe on a green field .
A lot of people riding ski lifts and standing on a ski slope .
A cow stands by parked motorcycles on a commercial street

walk 0.99
walk on 0.84
run 0.78
walk by 0.73
run on 0.61

A man in glasses walks through an open door .
A dog that is walking on a beach .
Several zebras running together in a field during the day .
People walking by a wall with graffiti on it and a billboard .
Three people are running on an ice rink .

The bottom half of a tennis player holding a racket.

hold 0.98
grip 0.91
hold up 0.84
grasp 0.79
hold on 0.77

A man holding a dog shields his eyes while outside
A baseball player grips the bat and waits for the ball .
People hold up shots for a toast in a crowded kitchen .
Woman on tennis court grasping racket with both hands .
A person in blue holds a tennis racket on a court .

A horse walking on a road near water.

walk on 0.95
walk 0.84
step up 0.76
pace in 0.73
walk by 0.70

A bunch of sheep walking on a big grass terrain
Two men walking down the road across from each other .
Elephant stepping up to a porch outside in a village .
Two young zebras pace in their zoo enclosure
A brown bear walking by a metal fence

Figure 3: The visualization of action tag candidates ($\mathcal{M}^i$ and $\mathcal{M}^t$) and memory response texts ($\mathcal{R}^i$ and $\mathcal{R}^t$) (see Sec. 3.2 Memory Search). The words corresponding to action tags are highlighted in red.

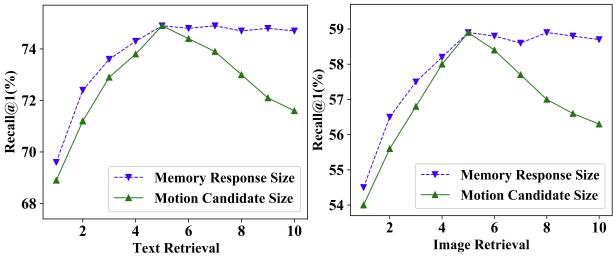


Figure 4: The variance of text retrieval R@1 (left) and image retrieval R@1 (right) in our method with different action tag candidate sizes and memory response sizes.

formances drop 4.8 % and 4.6 % in terms of text retrieval and image retrieval R@1. Besides, the action score is also very helpful (row 1 v.s. row2), which indicates that action score is valuable for complementing the action information in embedding representation. Further, if we remove both the action score and the memory bank, the performance of our model is worse than most of recent methods, which also implies that the performance gain is mainly from the action score and the action-aware memory bank.

Effect of Action Tag Candidate Size and Memory Response Size We fix the memory response size n_r and action tag candidate size n_m as 5 in turn, and change the other one in the range of [1, 10] to plot the performance variance in Figure 4. As the action tag candidate size increases, the model performance on both evaluation metrics first increases and then decreases. This might be because when the action tag candidate size is large, more irrelevant action tags would also be selected as predicted actions, which degrades the quality of action-aware embeddings. Whereas, as the memory response size increases, the model performance first increases and then becomes stable, which indicates that the quality of action-aware embedding will get stable when the number of action-similar texts is large enough. To balance the retrieval efficiency and performance, we choose 5 as the action tag candidate size.

	Action Score	Memory Bank	R@1(T)	R@1(I)	mR
1	✓	✓	74.9	58.9	83.2
2	×	✓	73.8	58.2	82.4
3	✓	×	70.1	54.3	80.1
4	×	×	69.2	52.9	79.1

Table 4: The ablation study of action score and memory bank. ✓ (*resp.*, ×) means adding (*resp.*, removing) the memory bank or the action score. T and I are short for text and image retrieval, respecticvely. When memory bank is removed, the triplet loss in Eqn. 6 is employed to $\mathbf{i}^e$ and $\mathbf{t}^e$.

4.4 Visualization of Action Tags and Memory Response Texts

In Figure 3, we visualize the confidence score of action candidates ($\mathcal{M}_i$ and $\mathcal{M}_t$) in a decreasing order. We also show memory response texts ($\mathcal{R}_i$ and $\mathcal{R}_t$). For query texts, the predicted action tags are mainly the verb phrases appearing in the text and their synonyms. For query images, the predicted action tags contain the potential actions from different aspects, like "hold" and "hold up" from the hand over racket and the "stand" and "stand by" from the foot over ground. Besides, we also observe that the memory response texts include some noise. However, as mentioned in Sec. 3.2, the transformer encoder E^m can prevent the noise in memory response features from being directly introduced to action-aware embeddings. More visualization and analyses of retrieval cases can be found in Supplementary.

5 Conclusion

In this paper, we have studied image-text retrieval from a new viewpoint, *i.e.*, enhancing embedding representation via action-aware information. We have proposed a novel method for action-aware embedding enhancement, with retrieval performed in both fragment-level embedding space and action-aware embedding space. Comprehensive experiments on two large-scale benchmark datasets have demonstrated that our method significantly outperforms the state-of-the-art approaches.

Acknowledgements

This work was supported by the NSF of China (Grant No. 62076162), the Shanghai Municipal Science and Technology Major Project (Grant No. 2021SHZDZX0102), and Shanghai Municipal Science and Technology Key Project (Grant No. 20511100300). We thank Wu Wen Jun Honorary Doctoral Scholarship, AI Institute, Shanghai Jiao Tong University.

References

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

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Lawrence Zitnick, C.; and Parikh, D. 2015. VQA: Visual question answering. In *ICCV*.

Chen, H.; Ding, G.; Lin, Z.; Zhao, S.; and Han, J. 2019. Cross-modal image-text retrieval with semantic consistency. In *ACM MM*.

Chen, H.; Ding, G.; Liu, X.; Lin, Z.; Liu, J.; and Han, J. 2020a. IMRAM: Iterative matching with recurrent attention memory for cross-modal image-text retrieval. In *CVPR*.

Chen, T.; and Luo, J. 2020. Expressing objects just like words: Recurrent visual embedding for image-text matching. In *AAAI*.

Chen, Y.; Cao, Y.; Hu, H.; and Wang, L. 2020b. Memory enhanced global-local aggregation for video object detection. In *CVPR*.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NeurIPS Workshop*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Faghri, F.; Fleet, D. J.; Kiros, J. R.; and Fidler, S. 2018. VSE++: Improving visual-semantic embeddings with hard negatives. In *BMVC*.

Frome, A.; Corrado, G. S.; Shlens, J.; Bengio, S.; Dean, J.; Ranzato, M.; and Mikolov, T. 2013. DeViSE: A deep visual-semantic embedding model. In *NeurIPS*.

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

Huang, Y.; and Wang, L. 2019. ACMM: Aligned Cross-Modal Memory for Few-Shot Image and Sentence Matching. In *ICCV*.

Huang, Y.; Wang, W.; and Wang, L. 2017. Instance-aware image and sentence matching with selective multimodal LSTM. In *CVPR*.

Ji, Z.; Lin, Z.; Wang, H.; and He, Y. 2020. Multi-modal memory enhancement attention network for image-text matching. *IEEE Access*, 8: 38438–38447.

Karpathy, A.; Joulin, A.; and Fei-Fei, L. 2014. Deep fragment embeddings for bidirectional image sentence mapping. In *NeurIPS*.

Kiros, R.; Salakhutdinov, R.; and Zemel, R. S. 2014. Unifying visual-semantic embeddings with multimodal neural language models. In *NeurIPS*.

Lee, K.-H.; Chen, X.; Hua, G.; Hu, H.; and He, X. 2018. Stacked cross attention for image-text matching. In *ECCV*.

Li, J.; Liu, L.; Niu, L.; and Zhang, L. 2021. Memorize, Associate and Match: Embedding Enhancement via Fine-Grained Alignment for Image-Text Retrieval. *IEEE Transactions on Image Processing*, 30: 9193–9207.

Li, K.; Zhang, Y.; Li, K.; Li, Y.; and Fu, Y. 2019. Visual semantic reasoning for image-text matching. In *ICCV*.

Li, S.; Xiao, T.; Li, H.; Yang, W.; and Wang, X. 2017. Identity-aware textual-visual matching with latent co-attention. In *ICCV*.

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

Liu, C.; Mao, Z.; Liu, A.-A.; Zhang, T.; Wang, B.; and Zhang, Y. 2019. Focus your attention: A bidirectional focal attention network for image-text matching. In *ACM MM*.

Liu, C.; Mao, Z.; Zhang, T.; Xie, H.; Wang, B.; and Zhang, Y. 2020. Graph structured network for image-text matching. In *CVPR*.

Ma, L.; Lu, Z.; Shang, L.; and Li, H. 2015. Multimodal convolutional neural networks for matching image and sentence. In *Proceedings of the IEEE international conference on computer vision*, 2623–2631.

Manning, C. D.; Surdeanu, M.; Bauer, J.; Finkel, J. R.; Bethard, S.; and McClosky, D. 2014. The Stanford CoreNLP natural language processing toolkit. In *ACL System Demonstrations*.

Miller, A.; Fisch, A.; Dodge, J.; Karimi, A.-H.; Bordes, A.; and Weston, J. 2016. Key-value memory networks for directly reading documents. In *EMNLP*.

Nam, H.; Ha, J.-W.; and Kim, J. 2017. Dual attention networks for multimodal reasoning and matching. In *CVPR*.

Niu, Z.; Zhou, M.; Wang, L.; Gao, X.; and Hua, G. 2017. Hierarchical multimodal LSTM for dense visual-semantic embedding. In *ICCV*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. GloVe: Global Vectors for Word Representation. In *EMNLP*.

Qu, L.; Liu, M.; Cao, D.; Nie, L.; and Tian, Q. 2020. Context-Aware Multi-View Summarization Network for Image-Text Matching. In *ACM MM*.

Qu, L.; Liu, M.; Wu, J.; Gao, Z.; and Nie, L. 2021. Dynamic Modality Interaction Modeling for Image-Text Retrieval. In *SIGIR*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*.

Song, G.; Wang, D.; and Tan, X. 2018. Deep memory network for cross-modal retrieval. *IEEE Transactions on Multimedia*, 21(5): 1261–1275.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*.

Vendrov, I.; Kiros, R.; Fidler, S.; and Urtasun, R. 2016. Order-embeddings of images and language. In *ICLR*.

Vinyals, O.; Toshev, A.; Bengio, S.; and Erhan, D. 2015. Show and tell: A neural image caption generator. In *CVPR*.

Wang, H.; Zhang, Y.; Ji, Z.; Pang, Y.; and Ma, L. 2020a. Consensus-Aware Visual-Semantic Embedding for Image-Text Matching. In *ECCV*.

Wang, L.; Li, Y.; and Lazebnik, S. 2016. Learning deep structure-preserving image-text embeddings. In *CVPR*.

Wang, M.; Lu, Z.; Li, H.; and Liu, Q. 2016. Memory-enhanced decoder for neural machine translation. In *EMNLP*.

Wang, S.; Chen, Y.; Zhuo, J.; Huang, Q.; and Tian, Q. 2018. Joint global and co-attentive representation learning for image-sentence retrieval. In *ACM MM*.

Wang, S.; Wang, R.; Yao, Z.; Shan, S.; and Chen, X. 2020b. Cross-modal scene graph matching for relationship-aware image-text retrieval. In *WACV*.

Wang, T.; Xu, X.; Yang, Y.; Hanjalic, A.; Shen, H. T.; and Song, J. 2019a. Matching images and text with multi-modal tensor fusion and re-ranking. In *ACM MM*.

Wang, Y.; Yang, H.; Qian, X.; Ma, L.; Lu, J.; Li, B.; and Fan, X. 2019b. Position focused attention network for image-text matching. In *AAAI*.

Wang, Z.; Liu, X.; Li, H.; Sheng, L.; Yan, J.; Wang, X.; and Shao, J. 2019c. CAMP: Cross-modal adaptive message passing for text-image retrieval. In *ICCV*.

Wehrmann, J.; Kolling, C.; and Barros, R. C. 2020. Adaptive Cross-Modal Embeddings for Image-Text Alignment. In *AAAI*.

Wei, X.; Zhang, T.; Li, Y.; Zhang, Y.; and Wu, F. 2020. Multi-Modality Cross Attention Network for Image and Sentence Matching. In *CVPR*.

Weikuo, G.; Huang, H.; Kong, X.; and He, R. 2019. Learning disentangled representation for cross-modal retrieval with deep mutual information estimation. In *ACM MM*.

Weston, J.; Chopra, S.; and Bordes, A. 2014. Memory networks. arXiv:1410.3916.

Wu, Y.; Wang, S.; and Huang, Q. 2017. Online asymmetric similarity learning for cross-modal retrieval. In *CVPR*.

Wu, Y.; Wang, S.; Song, G.; and Huang, Q. 2019. Learning fragment self-attention embeddings for image-text matching. In *ACM MM*.

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

Zellers, R.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. From recognition to cognition: Visual commonsense reasoning. In *CVPR*.

Zhang, Y.; and Lu, H. 2018. Deep cross-modal projection learning for image-text matching. In *ECCV*.