Dual-path Convolutional Image-Text Embeddings with Instance Loss

ZHEDONG ZHENG, University of Technology Sydney
LIANG ZHENG, The Australian National University
MICHAEL GARRETT, CingleVue International Australia and Edith Cowan University, Australia
YI YANG, University of Technology Sydney
MINGLIANG XU, Zhengzhou University
YI-DONG SHEN, State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences

Matching images and sentences demands a fine understanding of both modalities. In this article, we propose a new system to discriminatively embed the image and text to a shared visual-textual space. In this field, most existing works apply the ranking loss to pull the positive image/text pairs close and push the negative pairs apart from each other. However, directly deploying the ranking loss on heterogeneous features (i.e., text and image features) is less effective, because it is hard to find appropriate triplets at the beginning. So the naive way of using the ranking loss may compromise the network from learning inter-modal relationship. To address this problem, we propose the instance loss, which explicitly considers the intra-modal data distribution. It is based on an unsupervised assumption that each image/text group can be viewed as a class. So the network can learn the fine granularity from every image/text group. The experiment shows that the instance loss offers better weight initialization for the ranking loss, so that more discriminative embeddings can be learned. Besides, existing works usually apply the off-the-shelf features, i.e., word2vec and fixed visual feature. So in a minor contribution, this article constructs an end-to-end dual-path convolutional network to learn the image and text representations. End-to-end learning allows the system to directly learn from the data and fully utilize the supervision. On two generic retrieval datasets (Flickr30k and MSCOCO), experiments demonstrate that our method yields competitive accuracy compared to state-of-the-art methods. Moreover, in language-based person retrieval, we improve the state of the art by a large margin. The code has been made publicly available.

CCS Concepts: • **Computing methodologies → Visual content-based indexing and retrieval**; **Image representations**;

Additional Key Words and Phrases: Image-sentence retrieval, cross-modal retrieval, language-based person search, convolutional neural networks

Authors' addresses: Z. Zheng and Y. Yang, Centre for Artificial Intelligence, University of Technology Sydney, 15 Broadway, Ultimo NSW 2007, Australia; emails: Zhedong.Zheng@student.uts.edu.au, Yi.Yang@uts.edu.au; L. Zheng, The Australian National University, Room N214, ANU Campus, Australia 2601; email: liang.zheng@anu.edu.au; M. Garrett, Cinglevue International Australia and Edith Cowan University, 270 Joondalup Dr, Joondalup WA 6027, Australia; email: michael.garrett@cinglevue.com; M. Xu, Zhengzhou University, 100 Kexue Ave, Zhongyuan District, Zhengzhou, Henan, China; email: iexumingliang@zzu.edu.cn; Y.-D. Shen, Institute of Software, Chinese Academy of Sciences, 4th Zhongguancun South Fourth Street, Haidian District, Beijing, China; email: ydshen@ios.ac.cn.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2020 Association for Computing Machinery.
1551-6857/2020/05-ART51 $15.00
https://doi.org/10.1145/3383184

ACM Reference format:
Zhedong Zheng, Liang Zheng, Michael Garrett, Yi Yang, Mingliang Xu, and Yi-Dong Shen. 2020. Dual-path Convolutional Image-Text Embeddings with Instance Loss. *ACM Trans. Multimedia Comput. Commun. Appl.* 16, 2, Article 51 (May 2020), 23 pages.
https://doi.org/10.1145/3383184

1 INTRODUCTION

Image and text both contain very rich semantics but reside in heterogeneous modalities. Comparing to information retrieval within the same modality, matching image-text poses extra critical challenges, i.e., mapping images and text onto one shared feature space. For example, a model needs to distinguish between the "black dog," "gray dog," and "two dogs" in the text, and understand the visual differences in images depicting "black dog," "gray dog," and "two dogs." In this article, given an unseen image (text) query, we aim to measure its semantic similarity with the text (image) instances in the database and retrieve the true matched texts (images) to the query. Considering the testing procedure, this task requires connecting the two modalities with robust representations. In the early times, some relatively small datasets were used, e.g., Wikipedia [55] and Pascal Sentence [54], which contain around 3,000 and 5,000 image-text pairs, respectively. In recent years, several large-scale datasets with more than 30,000 images, including MSCOCO [41] and Flickr30k [78], have been introduced. Each image in these datasets is annotated with around five sentences. These large datasets allow deep architectures to learn robust representations and provide challenging evaluation scenarios.

During the past few years, ranking loss is commonly used as the objective function [13, 31, 47, 51, 56, 66] for image-text representation learning. The ranking loss aims to make the distance between positive pairs smaller than that between negative pairs by a predefined margin. In image-text matching, every training pair contains a visual feature and a textual feature. The ranking loss focuses on the distance between the two modalities. Its potential drawback is that it does not explicitly consider the feature distribution in a single modality. For example, when using ranking loss during training which does not distinguish between the slight differences in images, then given two testing images with slightly different semantics, the model may output similar descriptors for the two images. This is clearly undesirable for image/text matching considering the extremely fine granularity of this task. In our experiment, we observe that using the ranking loss alone in end-to-end training may cause the network to be stuck in a local minimum.

What motivates us is the effectiveness of class labels in earlier years of cross-media retrieval [58, 64, 65, 69, 71]. In these works, the class labels are annotated manually and during testing, the aim is to retrieve image/text belonging to the same class to the query. In light of this early practice, this article explores the feasibility of "class labels" in image/text matching, which is an instance retrieval problem. Two differences exist between cross-media retrieval on the category level [69, 71] and on the instance level (considered in this article). First, the true matches are those with the same category, and those with the exact same content with the query, respectively. That is to say, instance-level retrieval has a more strict matching criteria than category-level retrieval. Second, instance-level retrieval does not assume the existence of class labels. In this field of research, only image/text pairs are utilized during training. Given the intrinsic differences between the two tasks, it is non-trivial to directly transfer the experience from using class labels in category-level retrieval to instance-level retrieval.

Without annotated class labels, how can we initiate the investigation of the underlying data structures in the image/text embedding space? In this article, we name an image and its associated sentences an "image/text group." Our key assumption is that each "image/text" group is different

Fig. 1. Motivation. We define an image/text group as an image with its associated sentences. We observe that an image/text group is more or less different from each other. Therefore, we view every image/text group as a distinct class during training, yielding the instance loss.

from the others and can be viewed as a distinct class (see Figure 1). So, we propose a classification loss called instance loss to classify the image/text groups. Using this unsupervised class labels as supervision, we aim to enforce the model to discriminate each two images and two sentences (from different groups). It helps to investigate the fine-grained difference in single modality (intra-modal) and provides a good initialization for ranking loss, which is a driving force for end-to-end retrieval representation learning. In more details, using such an unsupervised assumption, we train the network to classify every image/text group with the softmax loss. In the experiment, we show that the instance loss that classifies a large number of classes, i.e., 113,287 image/text groups on MSCOCO [41], is able to converge without any hyper-parameter tuning. Improved retrieval accuracy can be observed as a result of instance loss.

In addition, we notice in the field of image-text matching that most recent works employ off-the-shelf deep models for image feature extraction [21, 28, 33, 36, 42, 48, 51, 52, 56, 66, 67, 70]. The fine-tuning strategy commonly seen in other computer vision tasks [2, 80, 83] is rarely adopted. A drawback of using off-the-shelf models is that these models are usually trained to classify objects into semantic categories [19, 34, 59]. The classification models are likely to miss image details such as color, number, and environment, which may convey critical visual cues for matching images and texts. For example, a model trained on ImageNet [57] can correctly classify the three images as "dog"; but it may not tell the difference between *black dog* and *gray dog*, or between *one dog* and *two dogs*. The ability to convey critical visual cues is a necessary component in instance-level image-text matching. Similar observations have been reported with regards to image captioning [62]. Moreover, for the text feature, *word2vec* [49] is a popular choice in image-text matching [30, 33, 52, 66]. Aiming to model the context information, the *word2vec* model is learned through a shallow network to predict neighboring words. However, the *word2vec* model is trained on a large-scale news dataset, i.e., GoogleNews, which differs substantially from the text in the target dataset. We inspired by the practice in many computer vision tasks, i.e., using the model pre-trained on ImageNet for initialization. Instead of directly using the off-the-shelf *word2vec* embeddings, we explore the possibility of initializing the model weight with *word2vec* embedding and fine-tuning the weights using image-text matching datasets.

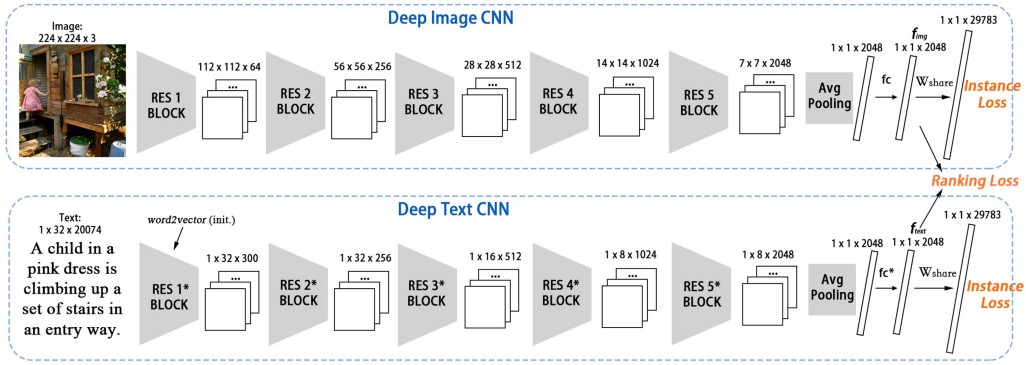

Fig. 2. We learn the image and text representations by two convolutional neural networks, i.e., deep image CNN (top) and deep text CNN (bottom). The deep image CNN is a ResNet-50 model [19] pre-trained on ImageNet. The deep text CNN is similar to the image CNN but with different basic blocks (see Figure 3). The text input is converted to the code of size $1 \times n \times d$, where n is the length of the sentence, and d denotes the size of the dictionary (more details could be found in Section 3.2). After the average pooling, we add one fully connected layer (input dim: 2,048, output dim: 2,048), one batchnorm layer, relu and one fully connected layer (input dim: 2,048, output dim: 2,048) in both image CNN and text CNN (We denote as fc and fc* in the figure, and the weights are not shared). Then, we add a shared-weight W_{share} classification layer (input dim: 2,048, output dim: 29,783). The objectives are the ranking loss and the proposed instance loss. On Flickr30k, for example, the model needs to classify 29,783 classes using instance loss.

Briefly, inspired by the effectiveness of class labels in early-time cross-media retrieval, we propose a similar practice in image-text matching called "instance loss." Instance loss works by providing better weight initialization for the ranking loss, thus producing more discriminative and robust image/text descriptions. Next, we also note that the pretrained CNN models may not meet the fine-grained requirement in image/text matching. So, we construct a dual-path CNN to extract image and text features directly from data rather. The network is end-to-end trainable and yields superior results to using features extracted from off-the-shelf models as input. Our contributions are summarized as follows:

- To provide better weight initialization and regularize the dual-path CNN model, we propose a large-number classification loss called instance loss. The robustness and effectiveness of instance loss are demonstrated by classifying each image/text group into one of the 113,287 classes on MSCOCO [41].
- We propose a dual-path CNN model for visual-textual embedding learning (see Figure 2). In contrast to the commonly used RNN+CNN model using fixed CNN features, the proposed CNN+CNN structure conducts efficient and effective end-to-end fine-tuning.
- We obtain competitive accuracy compared with the state-of-the-art image-text matching methods on three large-scale datasets, i.e., Flickr30k [78], MSCOCO [41], and CUHK-PEDES [39].

We note that Ma et al. also apply the CNN structure for text feature learning [47]. The main difference between our method and [47] is twofold. First, Ma et al. [47] use the ranking loss alone. In our method, we show that the proposed instance loss can further improve the result of ranking loss. Second, in Reference [47], four text CNN models are used to capture different semantic levels i.e., word, short phrase, long phrase and sentence. In this article, only one text CNN model is used and the word-level input is considered. Our model uses the residual block shown in Figure 3,

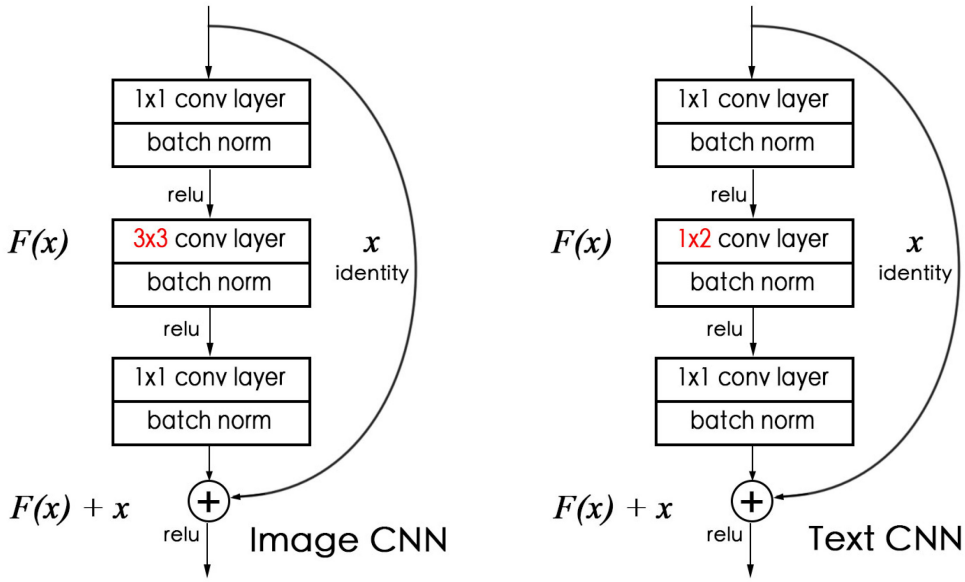

Fig. 3. The basic block of deep image CNN and deep text CNN. Similar with the local pattern of the images, the neighbor words in the sentence may contains important clues. The filter size in the image CNN is 3×3 with height and width padding; the filter size in the text CNN is 1×2 with length padding. Besides, we also use a shortcut connection, which helps to train a deep convolutional network [19]. The output $\mathcal{F}(x) + x$ has the same size with the input x.

which combines low level information i.e., word, as well as high-level inference to produce the final feature. In experiment (Tables 1 and 3), we show that using on the same image CNN (VGG-19), our method (with one text CNN) is superior to Reference [47] with text model ensembles by a large margin.

The rest of this article is organized as follows. Section 2 reviews and discusses the related works. Section 3 describes the proposed Image-Text CNN Structure in detail, followed by the objective function in Section 4. Training policy is described in Section 5. Experimental results and comparisons are discussed in Section 6 and conclusions are in Section 7. Furthermore, some qualitative results are included in Supplemental Material.

2 RELATED WORKS

The image-text bidirectional retrieval requires both understanding images and sentences in detail. In this section, we discuss some related works.

Deep models for image recognition. Deep models have achieved success in computer vision. The convolutional neural network (CNN) won the ILSVRC12 competition [57] by a large margin [34]. Later, VGGNet [59] and ResNet [19] further deepened the CNN and provide more insights into the network structure. In the field of image-text matching, most recent methods directly use fixed CNN features [21, 28, 33, 36, 42, 48, 51, 52, 56, 66, 67, 70] as input, which are extracted from the models pre-trained on ImageNet. While it is efficient to fix the CNN features and learn a visual-textual common space, it may lose the fine-grained differences between the images. This motivates us to fine-tune the image CNN branch in the image-text matching to provide for more discriminative embedding learning.

Deep models for natural language understanding. For natural language representation, *word2vec* [49] is commonly used [26, 30, 33, 52, 66]. This model contains two hidden layers, which learns from the context information. In the application of image-text matching, Klein et al. [33] and Wang et al. [66] pool word vectors extracted from the fixed *word2vec* model to form a sentence descriptor using Fisher vector encoding. Karpathy et al. [30] also utilize fixed word vectors as word-level input. With respect to this routine, this article proposes an equivalent scheme to fine-tuning the *word2vec* model, allowing the learned text representations to be adaptable to a specific task, which is, in our case, image-text matching.

Recurrent Neural Networks (RNN) are another common choice in natural language processing [50, 73]. Mao et al. [48], Cornia et al. [4], and Wang [63] employ a RNN to generate image captions with attention. Similarly, Nam et al. [51] utilize directional LSTM [23] for text encoding, yielding state-of-the-art multi-modal retrieval accuracy. Conversely, our approach is inspired by recent CNN breakthroughs on natural language understanding. For example, Gehring et al. apply CNNs to conduct machine translation, yielding competitive results and more than 9.3x speedup on the GPU [14]. There are also researchers who apply layer-by-layer CNNs for efficient text analysis [3, 25, 32, 81], obtaining competitive results in title recognition, event detection and text content matching. In this article, in place of RNNs, which are more commonly seen in image-text matching, we explore the usage of CNNs for text representation learning.

Multi-modal learning. There is a growing body of works on the interaction between multiple modalities. Some works focus on the efficient cross-modal searching by binary coding and hashing [6, 7, 37, 45, 64, 72, 75–77, 79, 85]. Others pay more attention to the effective retrieval by understanding the semantic meaning, which is close to this work. As for the content-based retrieval, one line of methods focus on **category-level retrieval** and leverage the category labels in the training set. Sharma et al. [58] extend the Canonical Correlation Analysis [18] (CCA) to learning class labels, and Wang et al. [65] learn the shared image-text space based on coupled input with class regression. Deng et al. propose a discriminative dictionary learning method [5]. Wu et al. [71] propose a bi-directional learning to rank for representation learning. In Reference [69], Wei et al. perform CNN fine-tuning by classifying categories on the training set and report an improved performance on image-text retrieval. Castrejon et al. deploy the multiple labels to learn the shared semantic space [1]. The second line of methods consider **instance-level retrieval** and, except for matched image-text pairs, do not provide any category label. Given a query, the retrieval objective is a specific image or related sentences [44]. Some works apply the auto-encoder to project high-dimensional features from different modalities onto a common low-dimensional latent space [9, 12, 68]. Some works deploy the pair-wise constraints. In Reference [20], He et al. use the assumption that the text and image components in a web document form a pairwise constraint. Zhang et al. consider the verification loss, using a binary classifier to classify the true matches and false matches [82]. Other works widely apply the ranking loss for instance-level retrieval [13, 31, 47, 51, 56, 66]. Karpathy et al. propose a part-to-part matching approach using a global ranking objective [31]. The "SPE" proposed in Reference [66] extends the ranking loss with structure-preserving constraints. SPE is similar to our work in that both works consider the intra-modal distance. Nevertheless, our work differs significantly from SPE. SPE enforces the model to rank the texts, i.e., considering the feature separability within the text modality only. In comparison, with the proposed instance loss, our method jointly discriminates the two modalities, i.e., images and their associated texts.

Briefly, we focus on instance-level retrieval and propose the instance loss, a novel contribution to the cross-modality community. It views each training image/text group as a distinct class and uses the softmax loss for model training. The assumption is unsupervised. We show that this method converges well and yields consistent improvement.

3 PROPOSED CNN STRUCTURE

In this article, we propose a dual-path CNN to simultaneously learn visual and textual representations in an end-to-end fashion, consisting of a deep image CNN for image input and one deep text CNN for sentence input. The entire network only contains four components, i.e., convolution, pooling, ReLU and batch normalisation. Compared to many previous methods that use off-the-shelf image CNNs [11, 21, 28, 33, 36, 42, 43, 48, 51, 52, 56, 66, 67, 70], end-to-end fine-tuning is superior in learning representations that encode image details (see Figure 2).

3.1 Deep Image CNN

We use ResNet-50 [19] pre-trained on ImageNet [34] as a basic model (the final 1,000-classification layer is removed) before conducting fine-tuning for visual feature learning. Given an input image of size 224×224, a forward pass of the network produces a 2,048-dimension feature vector. Followed by this feature, we add one fully connected layer (input dim: 2,048, output dim: 2,048), one batch normalization, relu and one fully connected layer (input dim: 2,048, output dim: 2,048). We denote the final 2,048-dim vector f_{img} as the visual descriptor of the input I. The forward pass process of the CNN, which is a non-linear function, is represented by function $\mathcal{F}_{img}(\cdot)$ defined as

$$f_{img} = \mathcal{F}_{img}(I). \tag{1}$$

3.2 Deep Text CNN

Text processing. Next, we describe our text processing method and the text CNN structure. Given a sentence, we first convert it into code T of size $n \times d$, where n is the length of the sentence, and d denotes the size of the dictionary. T is used as the input for the text CNN. We use *word2vec* [49] as a general dictionary to filter out rare words; if a word does not appear in the *word2vec* dictionary (3,000,000 words), it is discarded. For Flickr30k, we eventually use $d = 20{,}074$ words as the dictionary. Every word in Flickr30k thus can find an index $l \in [1, d]$ in the dictionary; for instance, a sentence of 18 words can be converted to $18 \times d$ matrix. The text input T can thus be formulated as:

$$T(i, j) = \begin{cases} 1 & \text{if } j = l_i \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

where $i \in [1, 18], j \in [1, d]$. The text CNN needs a fixed-length input. We set a fixed length 32 in this article, because about 98% of sentences contain less than 32 words. If the length of the sentence is shorter than 32, then we pad with zeros to the columns of T. If the length of the sentence is longer than 32, then we clip the final several words. Now, we obtain the $32 \times d$ sentence code T. We further reshape T into the $1 \times 32 \times d$ format, which can be considered as height, width and channel known in the image CNNs [19, 34].

Position shift. We are motivated by the jittering operation in the image CNN training. For text CNN, we apply a data augmentation policy called position shift. In a baseline approach, if the sentence length n is shorter than the standard input length 32, then a straightforward idea is to pad zeros at the end of the sentence, called *left alignment*. In the proposed position shift approach, we pad a random number of zeros at the beginning and the end of a sentence. In this manner, shift variations are contained in the text representation, so that the learned embeddings are more robust. In the experiment, we observe that position shift is of importance to the performance.

Deep text CNN. In the text CNN, filter size of the first convolution layer is $1 \times 1 \times d \times 300$, which can be viewed as a lookup table. Using the first convolutional layer, a sentence is converted to the word vector as follows. Given input T of $1 \times 32 \times d$, the first convolution layer results in a tensor of size $1 \times 32 \times 300$. There are two methods to initialize the first convolutional layer:

(1) random initialization [15], and (2) using the $d \times 300$-dim matrix from *word2vec* for initialization. In the experiment, we observe that *word2vec* initialization is superior to the random initialization.

For the rest of the text CNN, similar residual blocks are used as per the image CNN (see Figure 3). Similar to the local pattern in the image CNN, every two neighbor components may form a phrase containing content information. We set the filter size of convolution layers in basic text block to 1×2. Additionally, we add the shortcut connection in the basic block, which has been demonstrated to help training deep neural networks [19]. We apply basic blocks with a short connection to form the deep textual network (see Figure 2). The number of blocks is consistent with the ResNet-50 model in the visual branch. Given a sentence matrix T, its text descriptor f_{text} can be extract in an end-to-end manner from the text CNN $\mathcal{F}_{text}(\cdot)$:

$$f_{text} = \mathcal{F}_{text}(T). \tag{3}$$

4 PROPOSED INSTANCE LOSS

In this article, two types of losses are used, i.e., the standard ranking loss and the proposed instance loss. In Section 4.1, we briefly review the formulation of the ranking loss and discuss the limitation of the ranking loss. Section 4.2 describes the motivation and the formulation of the instance loss followed by a discussion. The differences between instance loss and ranking loss are discussed, and some primary experiments show the feasibility of instance loss. In Section 4.3, training convergence of the instance loss is discussed.

4.1 Ranking Loss Review

Ranking loss is a widely used objective function for retrieval problems. We use the cosine distance $D(f_{x_i}, f_{x_j}) = \frac{f_{x_i}}{||f_{x_i}||_2} \times \frac{f_{x_j}}{||f_{x_j}||_2}$ to measure the similarity between two samples, where f is the feature of a sample, and $||\cdot||_2$ denotes the L2-norm. The distance value $D(f_{x_i}, f_{x_j}) \in [-1, 1]$.

To effectively account for two modalities, we follow the ranking loss formulation as in some previous works [31, 51]. Here, I denotes the visual input, and T denotes the text input. Given a quadric input (I_a, T_a, I_n, T_n), where I_a, T_a describe the same image/text group, I_n, T_n are negative samples, ranking loss can be written as

$$L_{rank} = \overbrace{max[0, \alpha - (D(f_{I_a}, f_{T_a}) - D(f_{I_a}, f_{T_n}))]}^{image\ anchor} + \underbrace{max[0, \alpha - (D(f_{T_a}, f_{I_a}) - D(f_{T_a}, f_{I_n}))]}_{text\ anchor}, \tag{4}$$

where $D(\cdot, \cdot)$ is the cosine similarity, and α is a margin. Given an image query I_a, the similarity score of the correct text matching should be higher. Similarly, if we use sentence query T_a, then we expect the correct image content should be ranked higher. Ranking loss explicitly builds the relationship between the image and text.

Limitations of ranking loss. Although widely used, ranking loss has a potential drawback for the application of image-text matching. According to Equation (4), every pair contains a visual feature and a textual feature. The ranking loss focuses on the distance between the two modalities. So the potential drawback is that the ranking loss does not explicitly consider the feature distribution in a single modality. For instance, given two testing images with slightly different semantics, the model may output similar features. It is clearly undesirable for the extremely fine granularity of this task. In the experiment, using ranking loss alone is prone to get stuck in a local minimum (as to be shown in Figure 7 and Table 4).

Fig. 4. Sample images in the three datasets. For the MSCOCO and Flickr30k datasets, we view every image and its captions as an image/text group. For CUHK-PEDES, we view every identity (with several images and captions) as a class.

4.2 Instance Loss

Motivation. Some early works use coarse-grain category, i.e., art, biology, and sport, as the training supervision [58, 65, 69]. The multi-class classification loss has shown a good performance. But for instance-level retrieval, the classification loss has not been used. There may be two reasons. First, the category-level annotations are missing for most large-scale datasets. Second, if we use the category to train the model, then it forces different instances, i.e., black dog and white dogs, to the same class. It may compromise the CNN to learn the fine-grained difference.

In this article, we propose the instance loss for instance-level image-text matching. We define an image and its related text descriptions as an image/text group. In specific applications such as language-based person retrieval [38, 39], an image/text group is defined as images and their descriptions, which depict the same person (see Figure 4). Based on image/text groups, our assumption is that each image/text group is distinct (duplicates have been removed in the datasets). Under such assumption, we view each image/text group as a class. So, in essence, *instance loss is a softmax loss that classifies an image/text group into one of a large number of classes.* We want the trained model can tell the difference between every two images as well as every two sentences (from different groups). Formally, we define instance loss below.

Formulation. For two modalities, we formulate two classification objectives as follows:

$$P_{visual} = softmax\left(W_{share}^T f_{img}\right), \tag{5}$$

$$L_{visual} = -\log(P_{visual}(c)), \tag{6}$$

$$P_{textual} = softmax\left(W_{share}^T f_{text}\right), \tag{7}$$

$$L_{textual} = -\log(P_{text}(c)), \tag{8}$$

where f_{img} and f_{text} are image and text features defined in Equations (1) and (3), respectively. W_{share} is the parameter of the final fully connected layer (Figure 2). It can be viewed as concatenated weights $W_{share} = [W_1, W_2, \ldots, W_{29783}]$. Every weight W_i is a 2,048-dim vector. L denotes the loss and P denotes the probability over all classes. $P(c)$ is the predicted possibility of the right class c. **Here, we enforce shared weight W_{share} in the final fully connected layer for the two modalities, because otherwise the learned image and text features may exist in totally different subspaces.**

As to be described in Section 5, in the first training stage, the ranking loss is not used. We only use the instance loss; in the second training stage, both losses are used. The final loss function is

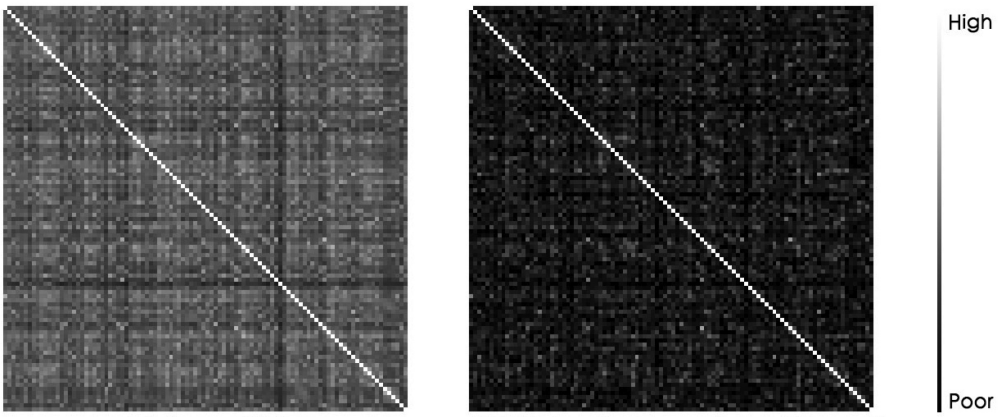

Fig. 5. We extract image features (2,048-dim) from a randomly selected 100 images in the Flickr30k validation set, using the ImageNet pre-trained ResNet-50 model and our model (after Stage I), respectively. We visualize the 100×100 Pearson's correlation. Lower Pearson's correlation between two features indicates higher orthogonality. The instance loss encourages the model to learn the difference between images.

a combination of the ranking loss and the instance loss, defined as

$$L = \lambda_1 L_{rank} + \lambda_2 L_{visual} + \lambda_3 L_{textual}, \tag{9}$$

where λ_1, λ_2, λ_3 are predefined weights for different losses.

Discussion. First, we show that instance loss provides better weight initialization than the ImageNet pretrained model. To prove this, we compare the image features from the off-the-shelf model pre-trained on ImageNet and the model trained with instance loss. Since the proposed instance loss explicitly considers the intra-modal distance, we observe that the feature correlation between two images is smaller after training with the instance loss (see Figure 5(b)). In fact, the instance loss encourages the model to find the fine-grained image details such as ball, stick, and frisbee to discriminate between image/text groups with similar semantics. We visualize the dog retrieval results in Figure 10. Our model can be well generalized to the test set and still sensitive to the subtle differences.

Second, we provide a two-class example to show the intuition of instance loss (Figure 6). After the convergence of Stage I, the instance loss pulls the data with the same label/group together, and pushes the data from different labels/groups away from each other. Although x_1 and y_1 are from different modality, the distance between x_1 and y_1 is closer, because they belong to the same group. In this manner, the positive pair (x_1, y_1) is closer than the negative pair (x_1, y_2). This property, as shown in the Figure 6 (*right*), will provide better weight initialization for the subsequent training using both the ranking loss and instance loss.

Third, we demonstrate that using the instance loss alone can lead to a decent initialization. To validate this point, we plot the distribution P of the intra-modal intra-class similarity $D_p = D(f_{x_i}, f_{y_i})$ and the distribution Q of the intra-modal inter-class similarity $D_n = D(f_{x_i}, f_{y_j})(j \neq i)$ on Flickr30k validation set (Figure 7(b)). We observe that, using instance loss alone, in most cases, leads to $D_p > D_n$ by a margin. The mean of D_p equals to 0.2405 while the mean of D_n is 0.0237.

Fourth, using the ranking loss alone achieves a relatively large margin between the positive pairs and negative pairs but there also exist many "hard" negative pairs (Figure 7(a)). These "hard" negative pairs usually have a high similarity, which compromises the matching performance of the

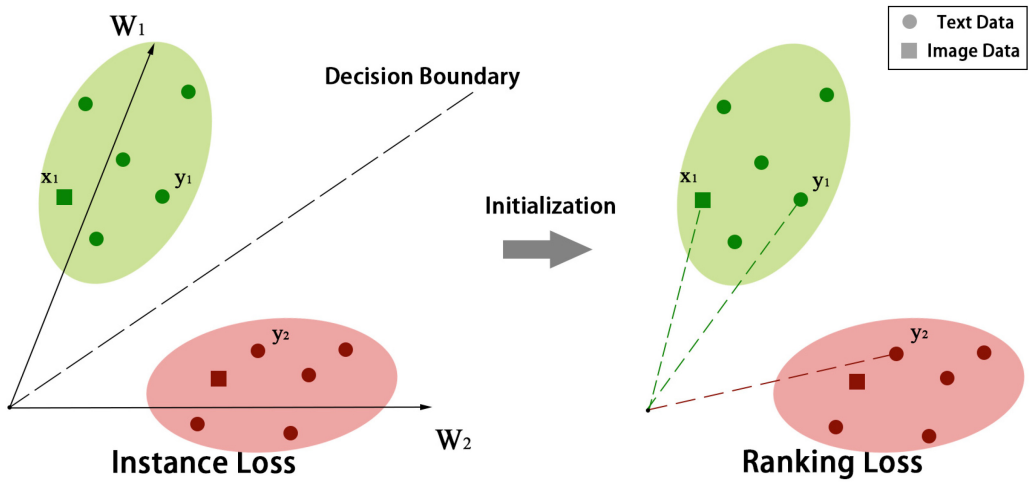

Fig. 6. Geometric Interpretation. Here, we give a two-class sample to show our intuition. The proposed instance loss pulls the samples with the same label together (close to either the relative weight W_1 or W_2). In this way, the positive pair (x_1, y_1) is closer than the negative pair (x_1, y_2). Stage I, therefore, leads to a decent weight initialization to be used in Stage II (ranking loss + instance loss).

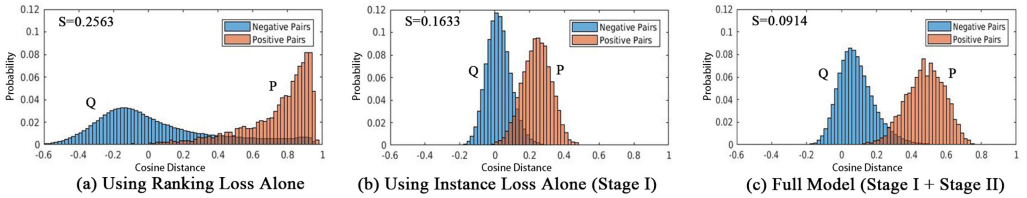

Fig. 7. The similarity (cosine distance) distribution of the positive pairs P and negative pairs Q on Flickr30k validation dataset. We show the result obtained by (a) using ranking loss alone, (b) using instance loss alone, and (c) full model (instance loss + ranking loss), respectively. Indicator S is calculated as the overlapping area between P and Q (defined in Section 4.2, lower is better). Through comparing their S values, the performance of the three methods is: "Full Model" > "Using Instance Loss Alone" > "Using Ranking Loss Alone."

true matches. To quantitatively compare the three models, we propose a simple indicator function,

$$S = \int_{-1}^{1} min(P(x), Q(x))dx, \tag{10}$$

which encodes the overlapping area of P and Q over the range of cosine similarity $[-1, 1]$. Indicator $S \in [0, 1]$. The smaller S is, the better the positive pairs and negative pairs are separated, and thus the better retrieval performance. $S = 1$ indicates the case where the two distributions, P and Q are completely overlapping. To the other extreme, $S = 0$ indicates that the positive pairs and negative pairs are perfectly separable: all the similarity scores of the positive pairs are larger than the similarity scores of the negative pairs. Therefore, a lower indicator score S indicates a better retrieval system.

In our experiment (Figure 7), the indicator scores of the three models are $S_{rank} = 0.2563$, $S_{instance} = 0.1633$ and $S_{full} = 0.0914$, respectively. It clearly demonstrates that in terms of the extent of feature separability: "Full Model" > "Using Instance Loss Alone" > "Using Ranking loss Alone." With the indicator function, we quantitatively show that using ranking loss alone pro-

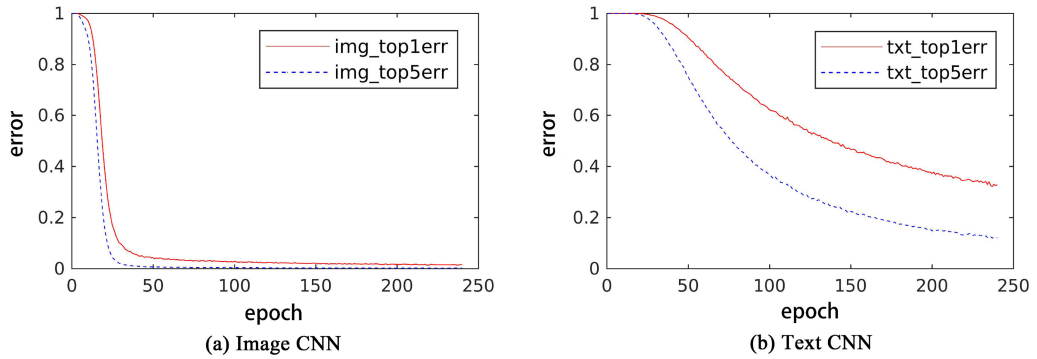

Fig. 8. Classification error curves when training on Flickr30k. The image CNN (a) and text CNN (b) converge well with 29,783 training classes (image/text groups).

duces more hard negative pairs than the proposed two competing methods, which compromises the matching performance of the ranking loss. In comparison, using instance loss alone produces a smaller S value, suggesting a better feature separability of the trained model. Importantly, when the two losses, i.e., ranking loss and instance loss, are combined, our full model has the smallest S value, indicating the fewest hard negative samples and the best retrieval accuracy among the three methods.

For the retrieval performance, using the instance loss alone can lead to a competitive accuracy in the experiment (Table 4). The effect of the instance loss is twofold. In the first training stage, when used alone, it pre-trains the text CNN and fine-tunes the two fully connected layers (and one batchnorm layer) of image CNN so that ranking loss can arrive at a better optimization for both modalities in the second stage (Figure 6). In the second training stage, when used together with ranking loss, it exhibits a regularization effect on the ranking loss.

4.3 Training Convergence of Instance Loss

The instance loss views every image/text group as a class, so the number of training classes is usually large. For instance, we have 29,783 classes when training on Flickr30k. In Figure 8, we show the training error curves of the image CNN and text CNN during training. We observe that the image CNN converges faster (Figure 8(a)), because the image CNN is pretrained on ImageNet. Text CNN converges more slowly, because most part of it is trained from scratch, but it still begins to learn something after 20 epochs, and finally converges after 240 epochs.

However, the convergence property is evidenced by some previous works. To our knowledge, some practices also suffer from limited data per class, because manually annotating data is usually expensive. For example, in person re-ID, CUHK03 dataset [40] has 9.6 training samples per class; VIPeR dataset [16] has 2 training samples per class. The previous works [53, 84] on CUHK03 and VIPeR show that the CNN classification model can be well trained as long as each class has more than a couple of training samples. In our case, there are usually six positive training samples per class (one image and five sentences). In the experiment, despite of the limited training data, the learned model has a good generalization ability on the validation set and test set, which accords with existing experience [53, 84].

5 A TWO-STAGE TRAINING PROCEDURE

We describe the training policy in this section. We split the training procedure into two stages. In the experiment, we show this policy helps the training.

Stage I: In this stage, we fix the pre-trained weights in the image CNN and use the proposed instance loss to tune the remaining part. The main reason is that most weights of the text CNN are learned from scratch. If we train the image and text CNNs simultaneously, then the text CNN may compromise the pre-trained image CNN. We only use the proposed instance loss in this stage ($\lambda_1 = 0, \lambda_2 = 1, \lambda_3 = 1$). It can provide a good initialization for the ranking loss. We note that even after Stage I, our network can achieve competitive results compared to previous works using off-the-shelf CNNs.

Stage II: After Stage I converges, we start Stage II for end-to-end fine-tuning of the *entire network*. Note that the weights of the image CNN are also fine-tuned. In this stage, we combine the instance loss with the ranking loss ($\lambda_1 = 1, \lambda_2 = 1, \lambda_3 = 1$), so that both classification and ranking errors are considered. In Section 6.4, we study the mechanism of the two losses. It can be observed that in Stage II, instance loss and ranking loss are complementary, thus further improving the retrieval result. Instance loss still regularizes the model and provides more attentions to discriminate the images and sentences. After Stage II (end-to-end fine-tuning), another round of performance improvement can be observed, and we achieve even more competitive performance.

6 EXPERIMENT

We first introduce the three large-scale image-text retrieval datasets, i.e., Flickr30k, MSCOCO, and CUHK-PEDES, followed by the evaluation metric in Section 6.1. Then Section 6.2 describes the implementation details and the reproducibility. We discuss the comparison with state of the art and mechanism study in Sections 6.3 and 6.4.

6.1 Datasets

Flickr30k [78] is one of the large-scale image captioning datasets. It contains 31,783 images collected from Flickr, in which every image is annotated with five text descriptions. The average sentence length is 10.5 words after removing rare words. We follow the protocol in References [24, 31] to split the dataset into 1,000 test images, 1,000 validation images, and 29,783 training images.

MSCOCO [41] contains 123,287 images and 616,767 descriptions. Every images contains roughly 5 text descriptions on average. The average length of captions is 8.7 after rare word removal. Following the protocol in Reference [30], we randomly select 5,000 images as test data and 5,000 images as validation data. The remaining 113,287 images are used as training data. The evaluation is reported on 1K test images (fivefold) and 5K test images.

CUHK-PEDES [39] collects images from many different person re-identification datasets. It contains 40,206 images from 13,003 different pedestrians and 80,440 descriptions. On average, each person has 3.1 images, and each image has two sentences. The average sentence length is 19.6 words after we remove rare words. We follow the protocol in [39], selecting the last 1,000 persons for evaluation. There are 3,074 test images with 6,156 captions, 3,078 validation images with 6,158 captions, and 34,054 training images with 68,126 captions.

Evaluation Metric We use two evaluation metrics, i.e., Recall@K and Median Rank. **Recall@K** is the possibility that the true match appears in the top K of the rank list, where a higher score is better. **Median Rank** is the median rank of the closest ground truth result in the rank list, with a lower index being better.

6.2 Implementation Details

The model is trained by stochastic gradient descent (SGD) with momentum fixed to 0.9 for weight update. While training, the images are resized to 224×224 pixels, which are randomly cropped from images whose shorter size is 256. We also perform simple data augmentation such as horizontal flipping. For training text input, we conduct position shift (Section 3.2) as data

augmentation. Dropout is applied to both CNNs, and the dropout rate is 0.75. For Flickr30k and MSCOCO, we set the max text length to 32; for CUHK-PEDES, we set the max text length to 56, since most sentences are longer. In the first training stage, we fixed the pre-trained image CNN, and train the text CNN only. The learning rate is 0.001. We stop training when instance loss converges. In the second stage, we combine the ranking loss as Equation (9) (the margin $\alpha = 1$) and fine-tune the entire network. When testing, we can use the trained image CNN and trained text CNN separately. We extract the image feature f_{img} by image CNN and the text feature f_{text} by text CNN. We use the cosine distance to evaluate the similarity between the query and candidate images/sentences. It is consistent with the similarity used in the ranking loss objective. The final retrieval result is based on the similarity ranking. We also conduct the horizontal flipping when testing and use the average features (no flip and flip) as the image feature.

Reproducibility. Our source code is available online.[1] The implementation is based on the Matconvnet package [60]. Since the entire network only uses four components, i.e., convolution, pooling, ReLU and batch normalization, it can be easily modified to other deep learning packages.

Training Time The image CNN (ResNet-50) in our method uses ~119 ms per image batch (batch size = 32) on an Nvidia 1080Ti GPU. The text CNN (similar ResNet-50) also uses ~117 ms per sentence batch (batch size = 32). Therefore, the image feature and text feature can be simultaneously calculated. Although our implementation is sequential, the model can run in a parallel style efficiently.

6.3 Comparison with State of the Art

We first compare our method with the state-of-the-art methods on the three datasets, i.e., Flickr30k, MSCOCO, and CUHK-PEDES. The compared methods include recent models on the bidirectional image and sentence retrieval. For a fair comparison, we present the results based on different image CNN structures, i.e., VGGNet [59] and ResNet [19]. We also summarise the visual and textual embeddings used in these works in Tables 1 and 3. Extensive results are shown in Tables 1, 3, and 2, respectively. On **Flickr30k**, we achieve competitive results with state-of-the-art DAN [51]: Recall@1 = 55.6%, Med r = 1 using image queries, and Recall@1 = 39.1%, Med r = 2 using text queries. While both based on VGG-19, our method exceeds DAN **6.2%** and **3.5%** Recall@1 using image and text query, respectively. On **MSCOCO** 1K-test-image setting, we arrive at Recall@1 = 65.6%, Med r = 1 using image queries, and Recall@1 = 47.1%, Med r = 2 using text queries. On 5K-test-image setting, we arrive at Recall@1 = 41.2%, Med r = 2 using image queries, and Recall@1 = 25.3%, Med r = 5 using text queries. CUHK-PEDES is a specific dataset for retrieving pedestrian images using the textual description. On **CUHK-PEDES**, we arrive at Recall@1 = 32.15%, Med r = 4. While both are based on a VGG-16 network, our model has **6.21%** higher recall rate. Moreover, our model based on ResNet-50 achieves new state-of-the-art performance: Recall@1 = 44.4%, Med r = 2 using language description to search relevant pedestrians. Our method exceeds the second best method [38] by **18.46%** in Recall@1 accuracy.

Note that m-CNN [47] also fine-tunes the CNN model to extract visual and textual features. m-CNN encompasses four different levels of text matching CNN, while we only use one deep textual model with residual blocks. While both are based on VGG-19, our model has higher performance than m-CNN. Compared with recent works, VSE++ [10], GXN [17] and CNP [29], our result is also competitive. SCAN [35] is published after our submission and achieves higher accuracy than us. The difference between the state-of-the-art method [35] and our method is provided below. First, Reference [35] uses a stronger visual feature extracted from the Faster RCNN. Second, Reference [35] applies a sequential text encoder, i.e., bi-directional GRU. It takes more training and test time

[1] https://github.com/layumi/Image-Text-Embedding.

Table 1. Method Comparisons on Flickr30k

Method	Visual	Textual	Image Query				Text Query			
			R@1	R@5	R@10	Med	R@1	R@5	R@10	Med r
DeVise [13]	ft AlexNet	ft skip-gram	4.5	18.1	29.2	26	6.7	21.9	32.7	25
DBRLM-7J7k [22]	fixed AlexNet	w2v	9.0	14.7	24.4	-	9.4	14.9	25.2	-
Deep Fragment [31]	ft RCNN	fixed word vector from [27]	16.4	40.2	54.7	8	10.3	31.4	44.5	13
DCCA [74]	ft AlexNet	TF-IDF	16.7	39.3	52.9	8	12.6	31.0	43.0	15
DVSA [30]	ft RCNN	w2v + ft RNN	22.2	48.2	61.4	4.8	15.2	37.7	50.5	9.2
LRCN [8]	ft VGG-16	ft RNN	23.6	46.6	58.3	7	17.5	40.3	50.8	9
m-CNN [47]	ft VGG-19	4 × ft CNN	33.6	64.1	74.9	3	26.2	56.3	69.6	4
VQA-A [42]	fixed VGG-19	ft RNN	33.9	62.5	74.5	-	24.9	52.6	64.8	-
GMM-FV [33]	fixed VGG-16	w2v + GMM + HGLMM	35.0	62.0	73.8	3	25.0	52.7	66.0	5
m-RNN [48]	fixed VGG-16	ft RNN	35.4	63.8	73.7	3	22.8	50.7	63.1	5
RNN-FV [36]	fixed VGG-19	feature from [33]	35.6	62.5	74.2	3	27.4	55.9	70.0	4
HM-LSTM [52]	fixed RCNN from [30]	w2v + ft RNN	38.1	-	76.5	3	27.7	-	68.8	4
SPE [66]	fixed VGG-19	w2v + HGLMM	40.3	68.9	79.9	-	29.7	60.1	72.1	-
sm-LSTM [28]	fixed VGG-19	ft RNN	42.5	71.9	81.5	2	30.2	60.4	72.3	3
RRF-Net [46]	fixed ResNet-152	w2v + HGLMM	47.6	77.4	87.1	-	35.4	68.3	79.9	-
2WayNet [9]	fixed VGG-16	feature from [33]	49.8	67.5	-	-	36.0	55.6	-	-
DAN (VGG-19) [51]	fixed VGG-19	ft RNN	41.4	73.5	82.5	2	31.8	61.7	72.5	3
VSE++ [10]	ft ResNet-152	ft RNN	52.9	-	87.2	1	39.6	-	79.5	2
DAN (ResNet-152) [51]	fixed ResNet-152	ft RNN	55.0	81.8	89.0	1	39.4	69.2	79.1	2
CNP [29]	fixed ResNet-152	ft RNN	55.5	82.0	89.3	-	41.1	70.5	80.1	-
GXN [17]	fixed ResNet-152	ft RNN	56.8	-	89.6	1	41.5	-	80.1	2
SCAN [35]	Faster R-CNN	ft RNN	67.9	90.3	95.8	-	48.6	77.7	85.2	-
Ours (VGG-19) Stage I	fixed VGG-19	ft ResNet-50† (w2v init.)	37.5	66.0	75.6	3	27.2	55.4	67.6	4
Ours (VGG-19) Stage II	ft VGG-19	ft ResNet-50† (w2v init.)	47.6	77.3	87.1	2	35.3	66.6	78.2	3
Ours (ResNet-50) Stage I	fixed ResNet-50	ft ResNet-50† (w2v init.)	41.2	69.7	78.9	2	28.6	56.2	67.8	4
Ours (ResNet-50) Stage II	ft ResNet-50	ft ResNet-50† (w2v init.)	53.9	80.9	89.9	1	39.2	69.8	80.8	2
Ours (ResNet-152) Stage I	fixed ResNet-152	ft ResNet-152† (w2v init.)	44.2	70.2	79.7	2	30.7	59.2	70.8	4
Ours (ResNet-152) Stage II	ft ResNet-152	ft ResNet-152† (w2v init.)	55.6	81.9	89.5	1	39.1	69.2	80.9	2

"Image Query" denotes using an image as query to search for the relevant sentences, and "Text Query" denotes using a sentence to find the relevant image. R@K is Recall@K (higher is better). Med r is the median rank (lower is better). "ft" means fine-tuning. †: Text CNN structure is similar to the image CNN, illustrated in Figure 3.

Table 2. Method Comparisons on CUHK-PEDES

Method	Visual	Text Query			
		R@1	R@5	R@10	Med r
CNN-RNN (VGG-16‡) [56]	fixed	8.07	-	32.47	-
Neural Talk (VGG-16‡) [61]	fixed	13.66	-	41.72	-
GNA-RNN (VGG-16‡) [39]	fixed	19.05	-	53.64	-
IATV (VGG-16) [38]	ft	25.94	-	60.48	-
Ours (VGG-16) Stage I	fixed	14.26	33.07	43.47	16
Ours (VGG-16) Stage II	ft	32.15	54.42	64.30	4
Ours (ResNet-50) Stage I	fixed	15.03	31.66	41.62	18
Ours (ResNet-50) Stage II	ft	44.40	66.26	75.07	**2**

R@K (%) is Recall@K (high is good). Med r is the median rank (low is good). ft means fine-tuning. ‡: pre-trained on person identification.

than the proposed text CNN. Text CNN does not depend on the output of the former words in the sentence and it is as fast as the Image CNN. In this work, we mainly seek to investigate the effect of the instance loss and ranking loss. The primary concern of our article is to prove that the instance loss + ranking loss model is superior to the commonly used ranking loss baseline (55.4% vs. 6.1%), and we report competitive performance.

Table 3. Method Comparisons on MSCOCO

Method	Visual	Textual	Image Query				Text Query			
			R@1	R@5	R@10	Med	R@1	R@5	R@10	Med r
1K test images										
DVSA [30]	ft RCNN	w2v + ft RNN	38.4	69.9	80.5	1	27.4	60.2	74.8	3
GMM-FV [33]	fixed VGG-16	w2v + GMM + HGLMM	39.4	67.9	80.9	2	25.1	59.8	76.6	4
m-RNN [48]	fixed VGG-16	ft RNN	41.0	73.0	83.5	2	29.0	42.2	77.0	3
RNN-FV [36]	fixed VGG-19	feature from [33]	41.5	72.0	82.9	2	29.2	64.7	80.4	3
m-CNN [47]	ft VGG-19	4 × ft CNN	42.8	73.1	84.1	2	32.6	68.6	82.8	3
HM-LSTM [52]	fixed CNN from [30]	ft RNN	43.9	-	87.8	2	36.1	-	86.7	3
SPE [66]	fixed VGG-19	w2v + HGLMM	50.1	79.7	89.2	-	39.6	75.2	86.9	-
VQA-A [42]	fixed VGG-19	ft RNN	50.5	80.1	89.7	-	37.0	70.9	82.9	-
sm-LSTM [28]	fixed VGG-19	ft RNN	53.2	83.1	91.5	1	40.7	75.8	87.4	2
2WayNet [9]	fixed VGG-16	feature from [33]	55.8	75.2	-	-	39.7	63.3	-	-
RRF-Net [46]	fixed ResNet-152	w2v + HGLMM	56.4	85.3	91.5	-	43.9	78.1	88.6	-
VSE++ [10]	ft ResNet-152	ft RNN	64.6	-	95.7	1	52.0	-	92.0	1
CNP [29]	fixed ResNet-152	ft RNN	69.9	92.9	97.5	-	56.7	87.5	94.8	-
GXN [17]	fixed ResNet-152	ft RNN	68.5	-	97.9	1	56.6	-	94.5	1
SCAN [35]	Faster R-CNN	ft RNN	72.7	94.8	98.4	-	58.8	88.4	94.8	-
Ours (VGG-19) Stage I	fixed VGG-19	ft ResNet-50† (w2v init.)	46.0	75.6	85.3	2	34.4	66.6	78.7	3
Ours (VGG-19) Stage II	ft VGG-19	ft ResNet-50† (w2v init.)	59.4	86.2	92.9	1	41.6	76.3	87.5	2
Ours (ResNet-50) Stage I	fixed ResNet-50	ft ResNet-50† (w2v init.)	52.2	80.4	88.7	1	37.2	69.5	80.6	2
Ours (ResNet-50) Stage II	ft ResNet-50	ft ResNet-50† (w2v init.)	65.6	89.8	95.5	1	47.1	79.9	90.0	2
5K test images										
GMM-FV [33]	fixed VGG-16	w2v + GMM + HGLMM	17.3	39.0	50.2	10	10.8	28.3	40.1	17
DVSA [30]	ft RCNN	w2v + ft RNN	16.5	39.2	52.0	9	10.7	29.6	42.2	14
VQA-A [42]	fixed VGG-19	ft RNN	23.5	50.7	63.6	-	16.7	40.5	53.8	-
VSE++ [10]	ft ResNet-152	ft RNN	41.3	-	81.2	2	30.3	-	72.4	4
GXN [17]	fixed ResNet-152	ft RNN	42.0	-	84.7	2	31.7	-	74.6	3
SCAN [35]	Faster R-CNN	ft RNN	50.4	82.2	90.0	-	38.6	69.3	80.4	-
Ours (VGG-19) Stage I	fixed VGG-19	ft ResNet-50† (w2v init.)	24.5	50.1	62.1	5	16.5	39.1	51.8	10
Ours (VGG-19) Stage II	ft VGG-19	ft ResNet-50† (w2v init.)	35.5	63.2	75.6	3	21.0	47.5	60.9	6
Ours (ResNet-50) Stage I	fixed ResNet-50	ft ResNet-50† (w2v init.)	28.6	56.2	68.0	4	18.7	42.4	55.1	8
Ours (ResNet-50) Stage II	ft ResNet-50	ft ResNet-50† (w2v init.)	41.2	70.5	81.1	2	25.3	53.4	66.4	5

R@K (%) is Recall@K (high is good). Med r is the median rank (low is good). 1K test images denotes using five non-overlap splits of 5K images to conduct retrieval evaluation and report the average result. 5K test images means using all images and texts to perform retrieval. ft means fine-tuning. †: Text CNN structure is similar to the image CNN, illustrated in Figure 3.

6.4 Mechanism Study

The effect of Stage I training. We replace the instance loss with the ranking loss at the first stage when fixing the image CNN. As shown in Table 4, the performance is limited. As discussed in Section 4.2, ranking loss focuses on inter-modal distance. It may be hard to tune the visual and textual features simultaneously at the beginning. As we expected, instance loss performs better, which focuses more on learning intra-modal discriminative descriptors. Besides, we observe that the result with both ranking loss and instance loss at Stage I is a bit lower than the one only using instance loss. We speculate that the ranking loss may not select good triplets at the beginning and converge to an inferior local minimum early, which also compromises the instance loss learning. Moreover, since the instance loss does not need the hard sample selection, the first stage using the instance loss is more computational efficient than using both losses.

Two losses can works together. In Stage II, the experiment on the validation set verifies that two losses can work together to improve the final retrieval result (see Table 4). Compared with models using only ranking loss or instance loss, the model with two losses provides for higher performance. In the second stage, instance loss does help to regularize the model.

Table 4. Ranking Loss and Instance Loss Retrieval Results on Flickr30k Validation Set

Method	Stage	Image Query		Text Query	
		R@1	R@10	R@1	R@10
Only Ranking Loss	I	6.1	27.3	4.9	27.8
Only Instance Loss	I	39.9	79.1	28.2	67.9
Instance Loss + Ranking Loss	I	37.6	75.1	24.1	65.6
Only Instance Loss	II	50.5	86.0	34.9	75.7
Only Ranking Loss	II	47.5	85.4	29.0	68.7
Full model	II	55.4	89.3	39.7	80.8

Except for the different losses, we apply the entirely same network (ResNet-50). For a clear comparison, we also fixed the image CNN in Stage I and tune the entire network in Stage II to observe the overfitting.

End-to-end fine-tuning helps. In Stage II, we fine-tune the entire network. For the two general object datasets Flickr30k and MSCOCO, fine-tuning the whole network can improve the rank-1 accuracy by approximately 10% (see Tables 1 and 3). Imagenet collects images from the Internet, while the pedestrian dataset CUHK-PEDES collects images from surveillance cameras. The fine-tuning result is more obvious on the CUHK-PEDES due to the different data distribution. The fine-tuned network (based on ResNet-50) improves the Recall@1 by 29.37%. The experiments indicate the end-to-end training is critical to image-sentence retrieval, especially person search.

Could we use one instance loss? One natural idea is to use one instance loss with the sum of two modality features. It could be formulated as

$$P_{both} = softmax\Big(W_{share}^{T}(f_{img} + f_{text})\Big), \tag{11}$$

$$Loss_{both} = -log(P_{both}). \tag{12}$$

We note that this loss does not equal to the proposed loss in Equations (5), (6), (7), and (8). The method using $f_{img} + f_{text}$ may depend on f_{img} or f_{text}. As shown in Table 8, the result is quite below the normal result on Flickr30k and MSCOCO. Because there is only one image sample for each class. It could be easy to cheat the loss and over-fit the image feature. The experiment shows that the f_{text} converges to small values, which are close to zero. The network is prone to only updating the f_{img} to cheat the loss. $P_{both} = softmax(W_{share}^{T} f_{img})$. Compared with one instance loss on $f_{img} + f_{text}$, the proposed two losses demand the network to learn f_{img} and f_{text} separately, which invades this unexpected condition.

Do we really need so many classes? For instance loss, the number of classes is usually large. Is it possible to use fewer classes? We implement the pseudo-category method by k-means clustering on MSCOCO, since MSCOCO has most images (classes). We use pool5 feature of ResNet50 pretrained on ImageNet to cluster 3,000 and 10,000 categories by K-means. The clustering results are used as the pseudo label for the images to conduct classification. Although clustering can decrease the number of training classes and add the samples per classes, different instances are forced to be of the same class and details may be lost (black/gray dog, two dogs), which compromises the accuracy. The retrieval result with k-classes on MSCOCO is shown in Table 5. It shows that the strategy is inferior to the instance loss.

Word2vec initialization helps. We compare the result using the *word2vec* initialization or random initialization [15] for the first convolution layer of text CNN. Note that we remove the words, which have not appeared in the training set, in the training data as well as dictionary. So the weight of first convolution layer is $d \times 300$ instead of $3{,}000{,}000 \times 300$. d is the dictionary size.

Table 5. K-class Loss vs. Instance Loss on MSCOCO

Methods	Image-Query R@1	Text-Query R@1
3,000 categories (StageI)	38.0	26.1
10,000 categories (StageI)	44.7	31.3
Our (StageI)	52.2	37.2

We use the K-means clustering result as pseudo categories. The experiment is based on Res50 + Res50† as the model structure.

Table 6. Ablation Study

Method	Image Query		Text Query	
	R@1	R@10	R@1	R@10
Random initialization [15]	38.0	78.7	26.6	66.6
Word2vec initialization	39.9	79.1	28.2	67.9

With/without word2vec initialization on Flickr30k validation. The result suggests *word2vec* serves as a proper initialization for text CNN.

Table 7. Ablation Study

Method	Image Query		Text Query	
	R@1	R@10	R@1	R@10
Left alignment	34.1	73.1	23.6	61.4
Position shift	39.9	79.1	28.2	67.9

Position shift vs. Left alignment on Flickr30k validation. It shows that position shift can serve as a significant data augmentation method for the text CNN.

Table 8. Ablation Study Using One Instance Loss on the $f_{img} + f_{text}$

Dataset	Image Query		Text Query	
	R@1	R@10	R@1	R@10
Flickr30k	0.3	1.7	0.1	1.4
MSCOCO	0.3	1.4	0.1	1.0

The result suggests that the method using $f_{img} + f_{text}$ may depend on f_{img} or f_{text}. The network is prone to overfit the dataset.

When testing, the missing words in the dictionary will also be removed in advance. As shown in Table 6, it can be observed that using *word2vec* initialization outperforms by 1% to 2% compared to the random initialization. Although *word2vec* is not trained on the target dataset, it still serves as a proper initialization for text CNN.

Position shift vs. Left alignment: Text CNN has a fixed-length input. As discussed in Section 3.2, left alignment is to pad zeros at the end of text input (like aligning the whole sentence left), if the length of the sentence is shorter than 32. Position shift is to add zeros at the end of text input as well as the beginning of the input. We conduct the position shift online when reading data from the disk. We do the experiment on Flickr30k validation set. As shown in Table 7, the model using position shift outperforms the one using left alignment ~5%. Position shift serves as a significant data augmentation method for text feature learning.

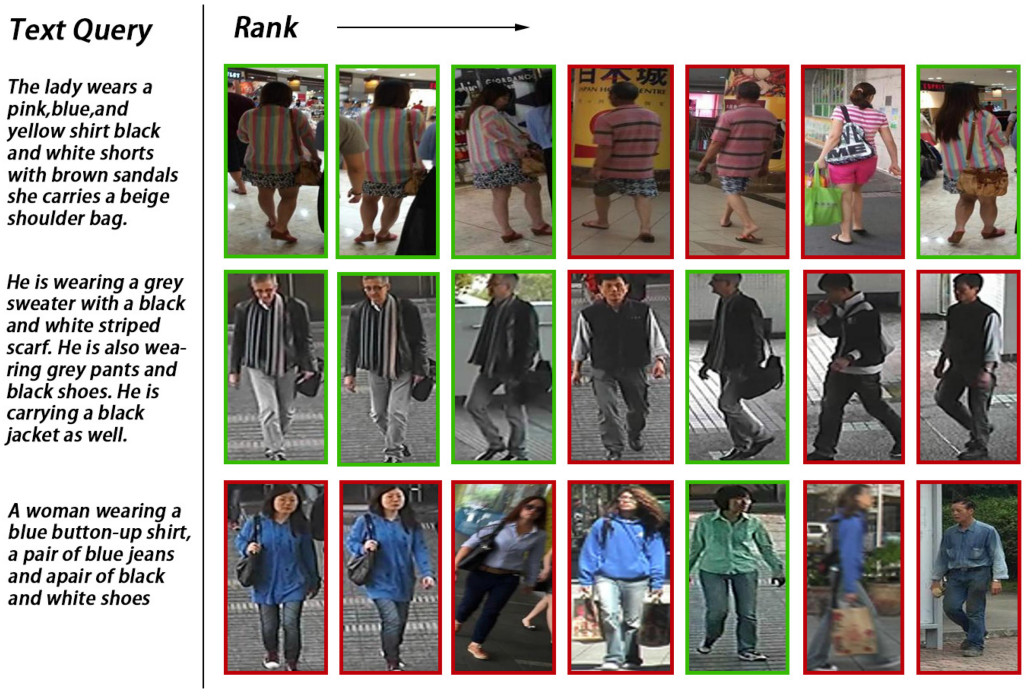

Fig. 9. Qualitative image search results using text query. The results are sorted from left to right according to their confidence. The images in green boxes are the true matches, and the images in red boxes are the false matches. In the last row, the rank-1 woman also wears a blue shirt, a pair of blue jeans and a pair of white shoes. The model outputs reasonable false matches.

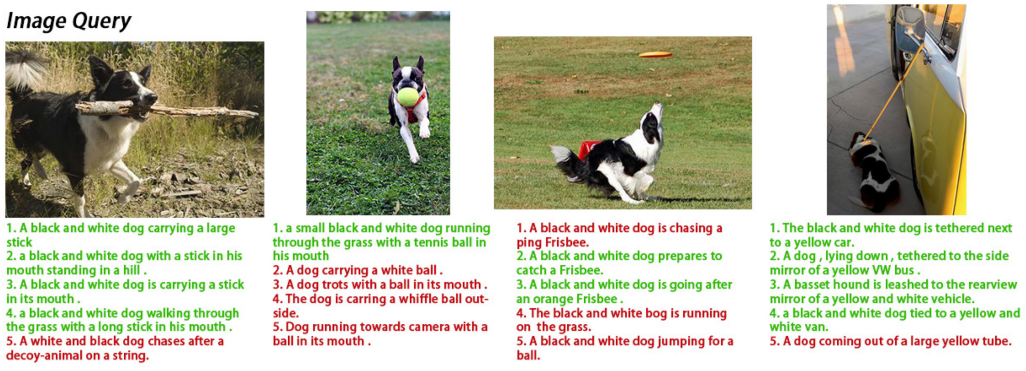

Fig. 10. Qualitative description search results using image query on Flickr30k. Below each image, we show the top five retrieval sentences (there are 5,000 candidate sentences in the gallery) in descending confidence. Here, we select four black and white dogs as our query. Except for the main object (dog), we show the model can correctly recognize environment and small object. The sentences in green are the true matches, and the descriptions in red are the false matches. Note that some general descriptions are also reasonable. (Best viewed when zoomed in.)

In Figures 9 and 10, we present some visual retrieval results on CUHK-PEDES and Flickr30k, respectively. Our method returns reasonable rank lists. (More qualitative results can be found in Supplemental Material.)

7 CONCLUSION

In this article, we propose the instance loss for image-text retrieval. It is based on an unsupervised assumption that every image/test group can be viewed as one class. The experiment shows instance loss can provide a proper initialization for ranking loss and further regularize the training. As a minor contribution, we propose a dual-path CNN to conduct end-to-end training on both image and text branches. The proposed method achieves competitive results on two generic retrieval datasets Flickr30k and MSCOCO. Furthermore, we arrive a +18% improvement on the person retrieval dataset CUHK-PEDES. Our code has been made publicly available. Additional examples can be found in Supplemental Material. We notice that there are limited training samples for each class, which may compromise the effectiveness of the proposed instance loss. In the future, we will investigate the feasibility of using generated samples for training. The generated samples could largely enrich the training set. However, the synthesis data may also introduce noise. How to use the generated text/images properly would be a new scientific problem. We will investigate high-fidelity sample generation and different pseudo-labeling methods.

REFERENCES

[1] Lluis Castrejon, Yusuf Aytar, Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2016. Learning aligned cross-modal representations from weakly aligned data. In *Proceedings of the CVPR*.

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. 2016. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv:1606.00915*.

[3] Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the ACL*.

[4] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. 2018. Paying more attention to saliency: Image captioning with saliency and context attention. *ACM Trans. Multimedia Comput. Commun. Appl.* 14, 2 (2018), 48.

[5] Cheng Deng, Xu Tang, Junchi Yan, Wei Liu, and Xinbo Gao. 2015. Discriminative dictionary learning with common label alignment for cross-modal retrieval. *IEEE Transactions on Multimedia* 18, 2 (2015), 208–218.

[6] Cheng Deng, Erkun Yang, Tongliang Liu, Wei Liu, Jie Li, and Dacheng Tao. 2019. Unsupervised semantic-preserving adversarial hashing for image search. *IEEE Trans. Image Process.* 28, 8 (2019), 4032–4044.

[7] Guiguang Ding, Yuchen Guo, Jile Zhou, and Yue Gao. 2016. Large-scale cross-modality search via collective matrix factorization hashing. *IEEE Transactions on Image Processing* 25, 11 (2016), 5427–5440.

[8] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the CVPR*.

[9] Aviv Eisenschtat and Lior Wolf. 2017. Linking image and text with 2-way nets. In *Proceedings of the CVPR*.

[10] Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2018. VSE++: Improved visual-semantic embeddings. In *Proceeding of BMVC* (2018).

[11] Hehe Fan, Liang Zheng, Chenggang Yan, and Yi Yang. 2018. Unsupervised person re-identification: Clustering and fine-tuning. *ACM Trans. Multimedia Comput. Commun. Appl.* 14, 4 (2018). DOI: https://doi.org/10.1145/3243316

[12] Fangxiang Feng, Xiaojie Wang, Ruifan Li, and Ibrar Ahmad. 2015. Correspondence autoencoders for cross-modal retrieval. *ACM Trans. Multimedia Comput. Commun. Appl.* 12, 1s (2015), 26.

[13] Andrea Frome, Greg S. Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. 2013. Devise: A deep visual-semantic embedding model. In *Proceedings of the NIPS*.

[14] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the ICML*.

[15] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the AISTAT*.

[16] Douglas Gray, Shane Brennan, and Hai Tao. 2007. Evaluating appearance models for recognition, reacquisition, and tracking. In *Proceedings of the PETS*.

[17] Jiuxiang Gu, Jianfei Cai, Shafiq R. Joty, Li Niu, and Gang Wang. 2018. Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models. In *Proceedings of the CVPR*.
[18] David R. Hardoon, Sandor Szedmak, and John Shawe-Taylor. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural Comput.* 16, 12 (2004), 2639–2664.
[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the CVPR*.
[20] Ran He, Man Zhang, Liang Wang, Ye Ji, and Qiyue Yin. 2015. Cross-modal subspace learning via pairwise constraints. *IEEE Transactions on Image Processing* 24, 12 (2015), 5543–5556.
[21] Xinwei He, Baoguang Shi, Xiang Bai, Gui-Song Xia, Zhaoxiang Zhang, and Weisheng Dong. 2019. Image caption generation with part of speech guidance. *Pattern Recogn. Lett.* 119 (2019), 229–237.
[22] Yonghao He, Shiming Xiang, Cuicui Kang, Jian Wang, and Chunhong Pan. 2016. Cross-modal retrieval via deep and bidirectional representation learning. *IEEE Trans. Multimedia* 18, 7 (2016), 1363–1377.
[23] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* 9, 8 (1997), 1735–1780.
[24] Micah Hodosh, Peter Young, and Julia Hockenmaier. 2013. Framing image description as a ranking task: Data, models and evaluation metrics. *J. Artific. Intell. Res.* 47 (2013), 853–899.
[25] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Proceedings of the NIPS*.
[26] Yuting Hu, Liang Zheng, Yi Yang, and Yongfeng Huang. 2018. Twitter100k: A real-world dataset for weakly supervised cross-media retrieval. *IEEE Transactions on Multimedia* 20, 4 (2017), 927–938.
[27] Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the ACL*.
[28] Yan Huang, Wei Wang, and Liang Wang. 2017. Instance-aware image and sentence matching with selective multimodal LSTM. In *Proceedings of the CVPR*.
[29] Yan Huang, Qi Wu, Chunfeng Song, and Liang Wang. 2018. Learning semantic concepts and order for image and sentence matching. In *Proceedings of the CVPR*.
[30] Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the CVPR*.
[31] Andrej Karpathy, Armand Joulin, and Fei Fei F. Li. 2014. Deep fragment embeddings for bidirectional image sentence mapping. In *Proceedings of the NIPS*.
[32] Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the EMNLP*.
[33] Benjamin Klein, Guy Lev, Gil Sadeh, and Lior Wolf. 2015. Associating neural word embeddings with deep image representations using fisher vectors. In *Proceedings of the CVPR*.
[34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Proceedings of the NIPS*.
[35] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. 2018. Stacked cross attention for image-text matching. In *Proceedings of the ECCV*.
[36] Guy Lev, Gil Sadeh, Benjamin Klein, and Lior Wolf. 2016. RNN fisher vectors for action recognition and image annotation. In *Proceedings of the ECCV*.
[37] Kai Li, Guo-Jun Qi, and Kien A. Hua. 2017. Learning label preserving binary codes for multimedia retrieval: A general approach. *ACM Trans. Multimedia Comput. Commun. Appl.* 14, 1 (2017), 2.
[38] 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 ICCV*.
[39] Shuang Li, Tong Xiao, Hongsheng Li, Bolei Zhou, Dayu Yue, and Xiaogang Wang. 2017. Person search with natural language description. In *Proceedings of the CVPR*.
[40] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. 2014. Deepreid: Deep filter pairing neural network for person re-identification. In *Proceedings of the CVPR*.
[41] 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 *Proceedings of the ECCV*.
[42] Xiao Lin and Devi Parikh. 2016. Leveraging visual question answering for image-caption ranking. In *Proceedings of the ECCV*.
[43] Yutian Lin, Liang Zheng, Zhedong Zheng, Yu Wu, Zhilan Hu, Chenggang Yan, and Yi Yang. 2019. Improving person re-identification by attribute and identity learning. *Pattern Recogn.* 95 (2019), 151–161. DOI: https://doi.org/10.1016/j.patcog.2019.06.006
[44] Ruoyu Liu, Yao Zhao, Shikui Wei, Liang Zheng, and Yi Yang. 2019. Modality-invariant image-text embedding for image-sentence matching. *ACM Trans. Multimedia Comput. Commun. Appl.* 15, 1 (2019), 1–19. DOI: https://doi.org/10.1145/3300939

[45] Xianglong Liu, Lei Huang, Cheng Deng, Bo Lang, and Dacheng Tao. 2016. Query-adaptive hash code ranking for large-scale multi-view visual search. *IEEE Transactions on Image Processing* 25, 10 (2016), 4514–4524.
[46] Yu Liu, Yanming Guo, Erwin M. Bakker, and Michael S. Lew. 2017. Learning a recurrent residual fusion network for multimodal matching. In *Proceedings of the ICCV*.
[47] Lin Ma, Zhengdong Lu, Lifeng Shang, and Hang Li. 2015. Multimodal convolutional neural networks for matching image and sentence. In *Proceedings of the ICCV*.
[48] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. 2015. Deep captioning with multimodal recurrent neural networks (m-rnn). In *Proceedings of the ICLR*.
[49] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv:1301.3781*.
[50] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of the Interspeech*.
[51] Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. 2017. Dual attention networks for multimodal reasoning and matching. In *Proceedings of the CVPR*.
[52] Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. 2017. Hierarchical multimodal LSTM for dense visual-semantic embedding. In *Proceedings of the ICCV*.
[53] Xuelin Qian, Yanwei Fu, Yu-Gang Jiang, Tao Xiang, and Xiangyang Xue. 2017. Multi-scale deep learning architectures for person re-identification. In *Proceedings of the CVPR*.
[54] Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier. 2010. Collecting image annotations using Amazon's mechanical turk. In *Proceedings of the NAACL HLT*. Association for Computational Linguistics, 139–147.
[55] 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 *Proceedings of the ACM MM*.
[56] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. 2016. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the CVPR*.
[57] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision* 115, 3 (2015), 211–252.
[58] Abhishek Sharma, Abhishek Kumar, Hal Daume, and David W. Jacobs. 2012. Generalized multiview analysis: A discriminative latent space. In *Proceedings of the CVPR*.
[59] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*.
[60] A. Vedaldi and K. Lenc. 2015. MatConvNet—Convolutional neural networks for MATLAB. In *Proceedings of the ACM MM*.
[61] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the CVPR*.
[62] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2017. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *Trans. Pattern Anal. Mach. Intell.* 39, 4 (2017), 652–663.
[63] Cheng Wang, Haojin Yang, and Christoph Meinel. 2018. Image captioning with deep bidirectional LSTMs and multi-task learning. *ACM Trans. Multimedia Comput. Commun. Appl.* 14, 2s (2018), 40.
[64] Di Wang, Xinbo Gao, Xiumei Wang, Lihuo He, and Bo Yuan. 2016. Multimodal discriminative binary embedding for large-scale cross-modal retrieval. *IEEE Transactions on Image Processing* 25, 10 (2016), 4540–4554.
[65] Kaiye Wang, Ran He, Wei Wang, Liang Wang, and Tieniu Tan. 2013. Learning coupled feature spaces for cross-modal matching. In *Proceedings of the ICCV*.
[66] Liwei Wang, Yin Li, and Svetlana Lazebnik. 2016. Learning deep structure-preserving image-text embeddings. In *Proceedings of the CVPR*.
[67] Liwei Wang, Yin Li, and Svetlana Lazebnik. 2017. Learning two-branch neural networks for image-text matching tasks. *arXiv:1704.03470*.
[68] Wei Wang, Beng Chin Ooi, Xiaoyan Yang, Dongxiang Zhang, and Yueting Zhuang. 2014. Effective multi-modal retrieval based on stacked auto-encoders. *Proc. VLDB Endow.* 7, 8 (2014), 649–660.
[69] Yunchao Wei, Yao Zhao, Canyi Lu, Shikui Wei, Luoqi Liu, Zhenfeng Zhu, and Shuicheng Yan. 2017. Cross-modal retrieval with cnn visual features: A new baseline. *IEEE Trans. Cybernet.* 47, 2 (2017), 449–460.
[70] Yunchao Wei, Yao Zhao, Zhenfeng Zhu, Shikui Wei, Yanhui Xiao, Jiashi Feng, and Shuicheng Yan. 2016. Modality-dependent cross-media retrieval. *ACM Trans. Intell. Syst. Technol.* 7, 4 (2016), 1–13.
[71] Fei Wu, Xinyan Lu, Zhongfei Zhang, Shuicheng Yan, Yong Rui, and Yueting Zhuang. 2013. Cross-media semantic representation via bi-directional learning to rank. In *Proceedings of the ACM MM*.
[72] Yu Wu, Yutian Lin, Xuanyi Dong, Yan Yan, Wei Bian, and Yi Yang. 2019. Progressive learning for person re-identification with one example. *IEEE Trans. Image Process.* 28, 6 (June 2019), 2872–2881. DOI: https://doi.org/10.1109/TIP.2019.2891895

[73] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv:1609.08144*.
[74] Fei Yan and Krystian Mikolajczyk. 2015. Deep correlation for matching images and text. In *Proceedings of the CVPR*.
[75] Yan Yan, Feiping Nie, Wen Li, Chenqiang Gao, Yi Yang, and Dong Xu. 2016. Image classification by cross-media active learning with privileged information. *IEEE Trans. Multimedia* 18, 12 (2016), 2494–2502.
[76] Erkun Yang, Cheng Deng, Chao Li, Wei Liu, Jie Li, and Dacheng Tao. 2018. Shared predictive cross-modal deep quantization. *IEEE Trans. Neural Netw. Learn. Syst.*99 (2018), 1–12.
[77] Yi Yang, Feiping Nie, Dong Xu, Jiebo Luo, Yueting Zhuang, and Yunhe Pan. 2011. A multimedia retrieval framework based on semi-supervised ranking and relevance feedback. *IEEE Trans. Pattern Anal. Mach. Intell.* 34, 4 (2011), 723–742.
[78] 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. *TACL* 2 (2014), 67–78.
[79] Changqing Zhang, Huazhu Fu, Qinghua Hu, Pengfei Zhu, and Xiaochun Cao. 2017. Flexible multi-view dimensionality co-reduction. *IEEE Transactions on Image Processing* 26, 2 (2016), 648–659.
[80] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. 2014. Part-based R-CNNs for fine-grained category detection. In *Proceedings of the ECCV*.
[81] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the NIPS*.
[82] Yuting Zhang, Luyao Yuan, Yijie Guo, Zhiyuan He, I-An Huang, and Honglak Lee. 2017. Discriminative bimodal networks for visual localization and detection with natural language queries. In *Proceedings of the CVPR*.
[83] Liang Zheng, Yi Yang, and Alexander G. Hauptmann. 2016. Person re-identification: Past, present, and future. *arXiv:1610.02984*.
[84] Zhedong Zheng, Liang Zheng, and Yi Yang. 2017. A discriminatively learned CNN embedding for person re-identification. *ACM Trans. Multimedia Comput. Commun. Appl.* 14, 1 (2017), 1–20. DOI: https://doi.org/10.1145/3159171
[85] Linchao Zhu, Zhongwen Xu, Yi Yang, and Alexander G. Hauptmann. 2017. Uncovering the temporal context for video question answering. *Int. J. Comput. Vision* 124, 3 (2017), 409–421. DOI: https://doi.org/10.1007/s11263-017-1033-7

Received August 2018; revised May 2019; accepted February 2020