Predicting Visual Features from Text for Image and Video Caption Retrieval

Jianfeng Dong, Xirong Li*, and Cees G. M. Snoek

***Abstract*—This paper strives to find amidst a set of sentences the one best describing the content of a given image or video. Different from existing works, which rely on a joint subspace for their image and video caption retrieval, we propose to do so in a visual space exclusively. Apart from this conceptual novelty, we contribute *Word2VisualVec*, a deep neural network architecture that learns to predict a visual feature representation from textual input. Example captions are encoded into a textual embedding based on multi-scale sentence vectorization and further transferred into a deep visual feature of choice via a simple multi-layer perceptron. We further generalize Word2VisualVec for video caption retrieval, by predicting from text both 3-D convolutional neural network features as well as a visual-audio representation. Experiments on Flickr8k, Flickr30k, the Microsoft Video Description dataset and the very recent NIST TrecVid challenge for video caption retrieval detail Word2VisualVec's properties, its benefit over textual embeddings, the potential for multimodal query composition and its state-of-the-art results.**

***Index Terms*—Image and video caption retrieval.**

I. Introduction

THIS paper attacks the problem of *image and video caption retrieval*, *i.e.,* finding amidst a set of possible sentences the one best describing the content of a given image or video. This challenging problem is often considered in tandem with its inverse: image and video retrieval from a sentence [1], [2], [3], [4], [5]. It is probably the reason why the prevailing image and video caption retrieval methods [6], [7], [8], [9], [10], [11] prefer to represent the visual and lingual modalities in a common latent subspace, before computing their similarities. Like others before us [12], [13], [14], we consider caption retrieval important enough by itself, but we question the dependence on latent subspace solutions. Our key novelty is that we find the most likely caption for a given image or video by looking for their similarity in the visual feature space exclusively, as illustrated in Fig. 1.

From the visual side we are inspired by the recent progress in predicting images from text [15], [16]. We also depart from the text, but instead of predicting pixels, our model predicts visual features. We consider features from deep convolutional neural networks (ConvNet) [17], [18], [19], [20], [21]. These neural networks learn a textual class prediction for an image by successive layers of convolutions, non-linearities, pooling, and full connections, with the aid of big amounts of labeled images, *e.g.,* ImageNet [22]. Apart from classification, visual features derived from the layers of these networks are superior representations for various challenges in vision [23], [24], [25], [26], [27] and multimedia [28], [29], [30], [31], [32]. We also rely on a layered neural network architecture, but rather than predicting a class label for an image, we strive to predict a deep visual feature from a natural language description for the purpose of caption retrieval.

From the lingual side we are inspired by the encouraging progress in sentence encoding by neural language modeling for cross-modal matching [33], [8], [34], [9], [10], [35]. In particular, word2vec [36] pre-trained on large-scale text corpora provides distributed word embeddings, an important prerequisite for vectorizing sentences towards a representation shared with image [33], [8] or video [11], [37]. In [9], [10], a sentence is fed as a word sequence into a recurrent neural network (RNN). The RNN output at the last time step is

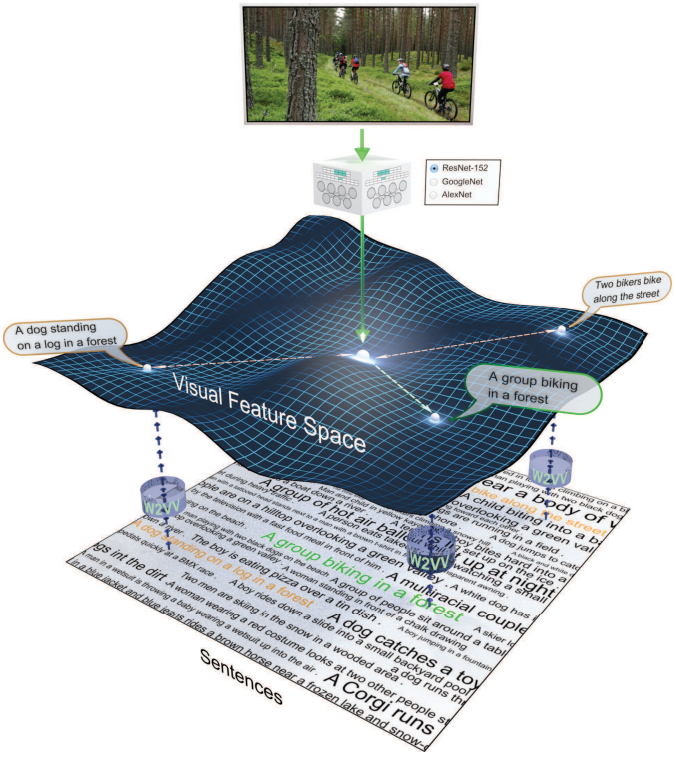

Fig. 1. **We propose to perform image and video caption retrieval in a visual feature space exclusively**. We achieve this by *Word2VisualVec* (W2VV), which predicts visual features from text.

*Corresponding author.

J. Dong is with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China (e-mail: danieljf24@zju.edu.cn).

X. Li is with the Key Lab of Data Engineering and Knowledge Engineering, School of Information, Renmin University of China, Beijing 100872, China (e-mail: xirong@ruc.edu.cn).

C. G. M. Snoek is with the Informatics Institute, University of Amsterdam, Amsterdam 1098 XH, The Netherlands, and also with Qualcomm Research, Amsterdam 1098 XH, The Netherlands (e-mail: cgmsnoek@uva.nl).

taken as the sentence feature, which is further projected into a latent subspace. We employ word2vec and RNN as part of our sentence encoding strategy as well. What is different is that we continue to transform the encoding into a higher-dimensional visual feature space via a multi-layer perceptron. As we predict visual features rather than latent features from text, we call our approach *Word2VisualVec*.

We make three contributions in this paper. First, to the best of our knowledge we are the first to solve the caption retrieval problem in the visual space only. Second, we propose Word2VisualVec, a deep neural network architecture that learns to predict a deep visual feature from an input sentence based on multi-scale sentence vectorization and a multi-layer perceptron. We consider prediction of several recent visual features [18], [20], [21] based on text, but the approach is general and can, in principle, predict any deep visual feature it is trained on. Third, we show how Word2VisualVec can be easily generalized to the video domain, by predicting from text both 3-D convolutional neural network features [38] as well as a visual-audio representation including Mel Frequency Cepstral Coefficients [39]. Experiments on Flickr8k [40], Flickr30k [41], the Microsoft Video Description dataset [42] and the very recent NIST TrecVid challenge for video caption retrieval [43] detail Word2VisualVec's properties, its benefit over the word2vec textual embedding, the potential for multimodal query composition and its state-of-the-art results. Before detailing our approach, we first highlight in more detail related work.

II. Related Work

A. Caption Retrieval

Prior to deep visual features, methods for image caption retrieval often resort to relatively complicated models to learn a shared representation, in order to compensate for the deficiency of traditional low-level visual features. Hodosh *et al.* [40] leverage Kernel CCA (Canonical Correlation Analysis), finding a joint embedding by maximizing the correlation between the projected image and text kernel matrices. With deep visual features, we observe an increased use of relatively light embeddings on the image side. Using the fc6 layer of a pre-trained AlexNet [17] as the image feature, Gong *et al.* show that linear CCA compares favorably to its kernel counterpart [6]. Linear CCA is also adopted by Klein *et al.* [8] for visual embedding. More recent models utilize affine transformations to reduce the image feature to a much shorter h-dimensional vector, with the transformation optimized in an end-to-end fashion within a deep learning framework [9], [10], [44].

Similar to the image domain, the state-of-the-art methods for video caption retrieval also operate in a shared subspace [11], [45], [4]. Xu *et al.* [11] propose to vectorize each subject-verb-object triplet extracted from a given sentence by a pre-trained word2vec, and subsequently aggregate the vectors into a sentence-level vector by a recursive neural network. A joint embedding model projects both the sentence vector and the video feature vector, obtained by temporal pooling over frame-level features, into a latent subspace. Otani *et al.* [45] improve upon [11] by exploiting web image search results of an input sentence, which are deemed helpful for word disambiguation, *e.g.,* telling if the word "keyboard" refers to a musical instrument or an input device for computers. To learn a common multimodal representation for videos and text, Yu *et al.* [4] use two distinct Long Short Term Memory (LSTM) modules to encode the video and text modalities respectively. They then employ a compact bilinear pooling layer to capture implicit interactions between the two modalities.

Different from the existing works, we propose to perform image and video caption retrieval directly in the visual space. This change is important as it allows us to completely remove the learning part from the visual side and focus our energy on learning an effective mapping from natural language text to the visual feature space.

B. Sentence Vectorization

To convert variably-sized sentences to fixed-sized feature vectors for subsequent learning, bag-of-words (BoW) is arguably the most popular choice [40], [6], [46], [47]. As a vocabulary has to be predefined based on the availability of words in training data, BoW cannot handle novel words outside the vocabulary. To overcome this limit, a distributional text embedding provided by word2vec [36] is gaining increased attention. The word embedding matrix used in [33], [11], [45], [48] is instantiated by a word2vec model pre-trained on large-scale text corpora. In Frome *et al.* [33], for instance, the input text is vectorized by averaging the word2vec vectors of its words. Such a mean pooling strategy results in a dense representation that could be less discriminative than the initial BoW feature. As an alternative, Klein *et al.* [8] and their follow-up [44] perform fisher vector pooling over word vectors.

Beside BoW and word2vec, we observe an increased use of RNN-based sentence vectorization. Socher *et al.* design a Dependency-Tree RNN that learns vector representations for sentences based on their dependency trees [34]. Lev *et al.* [49] propose RNN fisher vectors on the basis of [8], replacing the Gaussian model by a RNN model that takes into account the order of elements in the sequence. Kiros *et al.* [9] employ an LSTM to encode a sentence, using the LSTM's hidden state at the last time step as the sentence feature. In a follow-up work, Vendrov *et al.* replace LSTM by a Gated Recurrent Unit (GRU) which has less parameters to tune [10]. While RNN and its LSTM or GRU variants have demonstrated promising results for generating visual descriptions [50], [51], [52], [53], they tend to be over-sensitive to word orders by design. Indeed Socher *et al.* [34] suggest that for caption retrieval, models invariant to surface changes, such as word order, perform better.

In order to jointly exploit the merits of the BoW, word2vec and RNN based representations, we consider in this paper multi-scale sentence vectorization. Ma *et al.* [7] have made a first attempt in this direction. In their approach three multimodal ConvNets are trained on feature maps, formed by merging the image embedding vector with word, phrase and sentence embedding vectors. The relevance between an image

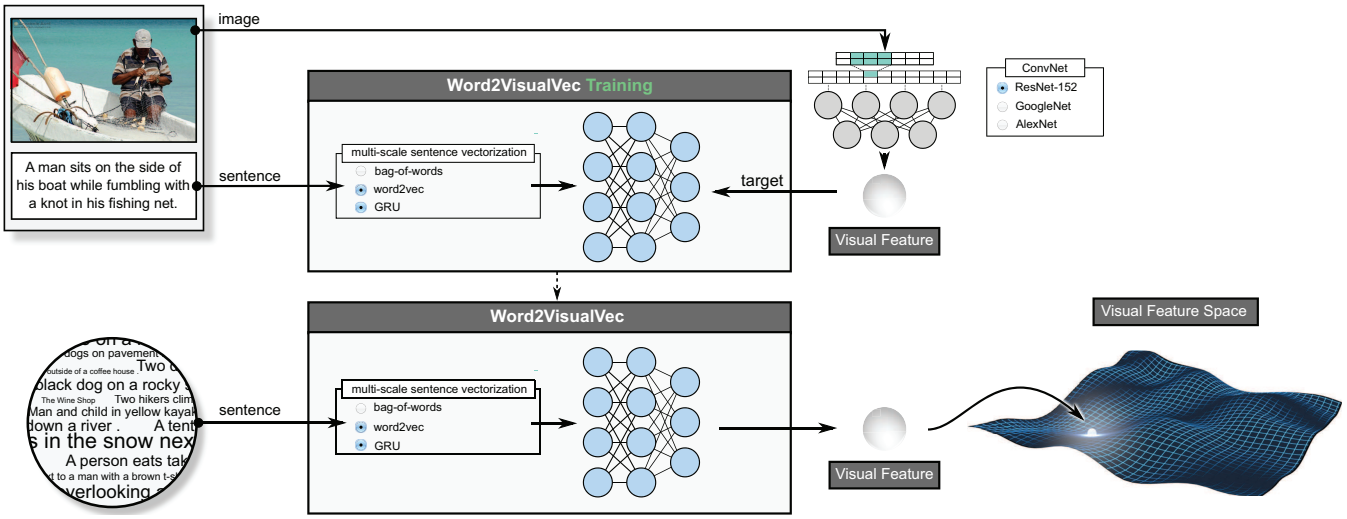

Fig. 2. **Word2VisualVec network architecture**. The model first vectorizes an input sentence into a fixed-length vector by relying on bag-of-words, word2vec and a GRU. The vector then goes through a multi-layer perceptron to produce the visual feature vector of choice, from a pre-trained ConvNet such as GoogleNet or ResNet. The network parameters are learned from image-sentence pairs in an end-to-end fashion, with the goal of reconstructing from the input sentence the visual feature vector of the image it is describing. We rely on the visual feature space for image and video caption retrieval.

and a sentence is estimated by late fusion of the individual matching scores. By contrast, we perform multi-scale sentence vectorization in an early stage, by merging BoW, word2vec and GRU sentence features and letting the model figure out the optimal way for combining them. Unlike [7], we do not require image-sentence pairs to generate feature maps. Our model predicts visual features from text alone, meaning the vectorization can be precomputed. An advantageous property for caption retrieval on large-scale image and video datasets.

III. Word2VisualVec

We propose to learn a mapping that projects a natural language description into a visual feature space. Consequently, the relevance between a given visual instance x and a specific sentence q can be directly computed in this space. More formally, let $\phi(x) \in \mathbb{R}^d$ be a d-dimensional visual feature vector. A pretrained ConvNet, apart from its original mission of visual class recognition, has now been recognized as an effective visual feature extractor [23]. We follow this good practice, instantiating $\phi(x)$ with a ConvNet feature vector. We aim for a sentence representation $r(q) \in \mathbb{R}^d$ such that the similarity can be expressed in terms of $\phi(x)$ and $r(q)$, say, in the form of an inner product. The proposed mapping model Word2VisualVec is designed to produce $r(q)$, as visualized in Fig. 2 and detailed next.

A. Architecture

Multi-scale sentence vectorization. To handle sentences of varying length, we choose to first vectorize each sentence. We propose multi-scale sentence vectorization that utilizes BoW, word2vec and RNN based text encodings.

BoW is a classical text encoding method. Each dimension in a BoW vector corresponds to the occurrence of a specific word in the input sentence, *i.e.,*

$$s_{bow}(q) = (c(w_1, q), c(w_2, q), \ldots, c(w_m, q)), \quad (1)$$

where $c(w, q)$ returns the occurrence of word w in q, and m is the size of a prespecified vocabulary. A drawback of BoW is that it cannot capture novel words outside the vocabulary. Given *faucet* as a novel word, for example, "A little girl plays with a faucet" will not have the main object encoded in its BoW vector. To compensate for such a loss, we further leverage word2vec. By learning from a large-scale text corpus, the vocabulary of word2vec is much larger than its BoW counterpart. We obtain the embedding vector of the sentence by mean pooling over its words, *i.e.,*

$$s_{word2vec}(q) := \frac{1}{|q|} \sum_{w \in q} v(w), \quad (2)$$

where $v(w)$ denotes individual word embedding vectors, $|q|$ is the sentence length. Previous works employ word2vec trained on web documents as their word embedding matrix [33], [50], [7]. However, recent studies suggest that word2vec trained on Flickr tags better captures visual relationships than its counterpart learned from web documents [54], [55]. We therefore train a 500-dimensional word2vec model on English tags of 30 million Flickr images, using the skip-gram algorithm [36]. This results in a vocabulary of 1.7 million words.

Despite their effectiveness, the BoW and word2vec representations ignore word orders in the input sentence. As such, they cannot discriminate between "a dog follows a person" and "a person follows a dog". To tackle this downside, we employ an RNN, which is known to be effective for modeling long-term word dependency in natural language text. In particular,

we adopt a GRU [56], which has less parameters than LSTM and presumably requires less amounts of training data. At a specific time step t, let v_t be the embedding vector of the t-th word, obtained by performing a lookup on a word embedding matrix W_e. GRU receives inputs from v_t and the previous hidden state h_{t-1}, and accordingly the new hidden state h_t is updated as follows,

$$\begin{aligned} z_t &= \sigma(W_{zv}v_t + W_{zh}h_{t-1} + b_z), \\ r_t &= \sigma(W_{rv}v_t + W_{rh}h_{t-1} + b_r), \\ \widetilde{h}_t &= tanh(W_{hv}v_t + W_{hh}(r_t \odot h_{t-1}) + b_h), \\ h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h}_t, \end{aligned} \tag{3}$$

where z_t and r_t denote the update and reset gates at time t respectively, while W and b with specific subscripts are weights and bias parameterizing the corresponding gates. The symbol $\odot$ indicates element-wise multiplication, while $\sigma(\cdot)$ is the sigmoid activation function. We re-use word2vec previously trained on the Flickr tags to initialize W_e. The last hidden state $h_{|q|}$ is taken as the RNN based representation of the sentence.

Multi-scale sentence vectorization is obtained by concatenating the three representations, that is

$$s(q) = [s_{bow}(q), s_{word2vec}(q), h_{|q|}]. \tag{4}$$

Text transformation via a multilayer perceptron. The sentence vector $s(q)$ goes through subsequent hidden layers until it reaches the output layer $r(q)$, which resides in the visual feature space. More concretely, by applying an affine transformation on $s(q)$, followed by an element-wise ReLU activation $\sigma(z) = \max(0, z)$, we obtain the first hidden layer $h_1(q)$ of an l-layer Word2VisualVec as:

$$h_1(q) = \sigma(W_1 s(q) + b_1). \tag{5}$$

The following hidden layers are expressed by:

$$h_i(q) = \sigma(W_i h_{i-1}(q) + b_i), i = 2, ..., l-2, \tag{6}$$

where W_i parameterizes the affine transformation of the i-th hidden layer and b_i is a bias terms. In a similar manner, we compute the output layer $r(q)$ as:

$$r(q) = \sigma(W_l h_{l-1}(q) + b_l). \tag{7}$$

Putting it all together, the learnable parameters are represented by $\theta = [W_e, W_{z.}, W_{r.}, W_{h.}, b_z, b_r, b_h, W_1, b_1, \ldots, W_l, b_l]$.

In principle, the learning capacity of our model grows as more layers are used. This also means more solutions exist which minimize the training loss, yet are suboptimal for unseen test data. We analyze in the experiments how deep Word2VisualVec can go without losing its generalization ability.

B. Learning algorithm

Objective function. For a given image, different persons might describe the same visual content with different words. For example, "A dog leaps over a log" versus "A dog is leaping over a fallen tree". The verb leap in different tenses essentially describe the same action, while a log and a fallen tree can have similar visual appearance. Projecting the two sentences into the same visual feature space has the effect of implicitly finding such correlations. In order to reconstruct the visual feature $\phi(x)$ directly from q, we use Mean Squared Error (MSE) as our objective function. We have also experimented with the marginal ranking loss, as commonly used in previous works [57], [58], [33], [59], but found MSE yields better performance.

The MSE loss l_{mse} for a given training pair is defined as:

$$l_{mse}(x, q; \theta) = (r(q) - \phi(x))^2. \tag{8}$$

We train Word2VisualVec to minimize the overall MSE loss on a given training set $\mathcal{D} = \{(x, q)\}$, containing a number of relevant image-sentence pairs:

$$\underset{\theta}{\operatorname{argmin}} \sum_{(x,q)\in\mathcal{D}} l_{mse}(x, q; \theta). \tag{9}$$

Optimization. We solve Eq. (9) using stochastic gradient descent with RMSprop [60]. This optimization algorithm divides the learning rate by an exponentially decaying average of squared gradients, to prevent the learning rate from effectively shrinking over time. We empirically set the initial learning rate $\eta = 0.0001$, decay weights $\gamma = 0.9$ and small constant $\epsilon = 10^{-6}$ for RMSprop. We apply dropout to all hidden layers in Word2VisualVec to mitigate model overfitting. Lastly, we take an empirical learning schedule as follows. Once the validation loss does not decrease in three consecutive epochs, we divide the learning rate by 2. Early stop occurs if the validation performance does not improve in ten consecutive epochs. The maximal number of epochs is 100.

C. Image Caption Retrieval

For a given image, we select from a given sentence pool the sentence deemed most relevant with respect to the image. Note that image-sentence pairs are required only for training Word2VisualVec. For a test sentence, its $r(q)$ is obtained by forward computation through the Word2VisualVec network, without the need of any test image. Hence, the sentence pool can be vectorized in advance. Image caption retrieval in our case boils down to finding the sentence nearest to the given image in the visual feature space. We use the cosine similarity between $r(q)$ and the image feature $\phi(x)$, as this similarity normalizes feature vectors and is found to be better than the dot product.

D. Video Caption Retrieval

Word2VisualVec is also applicable for video as long as we have an effective vectorized representation of video. Again, different from previous methods for video caption retrieval that execute in a joint subspace [11], [45], we project sentences into the video feature space.

Following the good practice of using pre-trained ConvNets for video content analysis [61], [25], [62], [63], we extract features by applying image ConvNets on individual frames and 3-D ConvNets [38] on consecutive-frame sequences. For short video clips, as used in our experiments, mean pooling over video frames is considered reasonable [61], [63]. Hence, the visual feature vector of each video is obtained by averaging

TABLE I
WHICH VISUAL FEATURE? FOR ALL THE FEATURES, WE USE A THREE-LAYER WORD2VISUALVEC OR WORD2VIDEOVEC. PREDICTING THE RESNET-152 FEATURE YIELDS THE BEST PERFORMANCE FOR BOTH IMAGES AND VIDEOS.

		Flickr8k			Flickr30k			MSVD		
ConvNet	**Layer**	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
CaffeNet	fc7	18.9	42.3	54.2	22.2	45.7	57.1	8.7	22.2	31.3
GoogLeNet	pool5	24.7	51.6	64.1	27.1	54.1	65.7	14.5	30.3	39.7
GoogLeNet-shuffle	pool5	30.2	57.6	70.5	33.3	63.5	73.5	**16.6**	33.7	43.4
ResNet-152	pool5	**32.1**	**62.9**	**75.5**	**36.5**	**65.0**	**75.1**	16.4	**34.8**	**46.6**
C3D	fc6	-	-	-	-	-	-	14.8	34.5	44.0

TABLE II
HOW TO VECTORIZE AN INPUT SENTENCE? MULTI-SCALE SENTENCE VECTORIZATION PROVIDES THE BEST PERFORMANCE ON FLICKR8K AND FLICKR30K, WHILE ITS SINGLE-SCALE COUNTERPART IS PREFERRED ON MSVD WHERE VISUAL EXAMPLES ARE IN SHORT SUPPLY.

	Flickr8k			Flickr30k			MSVD		
Sentence vectorization	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
BoW	34.7	62.9	74.7	41.8	70.9	78.6	15.8	32.1	39.9
word2vec	32.1	62.9	75.5	36.5	65.0	75.1	**16.4**	**34.8**	**46.6**
GRU	33.4	63.1	75.3	42.0	70.4	80.1	15.8	31.3	41.8
Multi-scale	**36.3**	**66.4**	**78.2**	**45.9**	**71.9**	**81.3**	16.1	34.5	43.1

the feature vectors of its frames. Note that longer videos open up possibilities for further improvement of Word2VisualVec by exploiting temporal order of video frames, *e.g.,* [64]. The audio channel of a video sometime provides complementary information to the visual channel. For instance, to help decide whether a person is talking or singing. To exploit this channel, we extract a bag of quantized Mel-frequency Cepstral Coefficients (MFCC) [39] and concatenate it with the previous visual feature. Word2VisualVec is trained to predict such a visual-audio feature, as a whole, from input text.

Word2VisualVec is used in a principled manner, transforming an input sentence to a video feature vector, let it be visual or visual-audio. For the sake of clarity we term the video variant *Word2VideoVec*.

IV. EXPERIMENTS

A. Properties of Word2VisualVec

We first investigate the impact of major design choices. Due to high complexity of the problem, evaluating all variables simultaneously is computationally prohibitive. The evaluation is thus conducted sequentially, focusing on one variable per time. For its efficient execution, word2vec is used for sentence vectorization in this experiment, unless stated otherwise.

Data. For image caption retrieval, we use two popular benchmark sets, Flickr8k [40] and Flickr30k [41]. Each image is associated with five crowd-sourced English sentences, which briefly describe the main objects and scenes present in the image. For video caption retrieval we rely on the Microsoft Video Description dataset (MSVD) [42]. Each video is labeled with 40 English sentences on average. The videos are short, usually less than 10 seconds long. For the ease of cross-paper comparison, we follow the identical data partitions as used in [59], [8], [10] for images and [61] for videos. That is, training / validation / test is 6k / 1k / 1k for Flickr8k, 29K / 1,014 / 1k for Flickr30k, and 1,200 / 100 / 670 for MSVD.

Evaluation criteria. Following the common convention [40], [7], [10], we report rank-based performance metrics $R@K$ $(K = 1, 5, 10)$. $R@K$ computes the percentage of test images for which at least one correct result is found among the top-K retrieved sentences. Hence, higher $R@K$ means better performance.

Which visual feature? A deep visual feature is determined by a specific ConvNet and its layers. We experiment with four pretrained 2-D ConvNets, *i.e.,* CaffeNet [18], GoogLeNet [20], GoogLeNet-shuffle [62] and ResNet-152 [21]. The first three 2-D ConvNets were trained using images containing 1K different visual objects as defined in the Large Scale Visual Recognition Challenge [22]. GoogLeNet-shuffle follows GoogLeNet's architecture, but is re-trained using a bottom-up reorganization of the complete 22K ImageNet hierarchy, excluding over-specific classes and classes with few images and thus making the final classes more balanced. For the video dataset, we further experiment with a 3-D ConvNet [38], trained on one million sports videos containing 487 sport-related concepts [65]. As the videos were muted, we cannot evaluate Word2VideoVec with audio features. We tried multiple layers of each ConvNet model and report the best performing layer. We see from Table I that as the ConvNets go deeper, predicting the corresponding visual features by Word2VisualVec improves. This result is encouraging as better performance can be expected from the continuous progress in deep learning features. In what follows we use the ResNet-152 feature because of its top performance.

How deep? We vary the number of MLP layers, and observe a performance peak when using three-layers, *i.e.,* 500-2048-2048, on Flickr8k and four-layers, *i.e.,* 500-2048-2048-2048, on Flickr30k. Recall that the model is chosen in terms of its

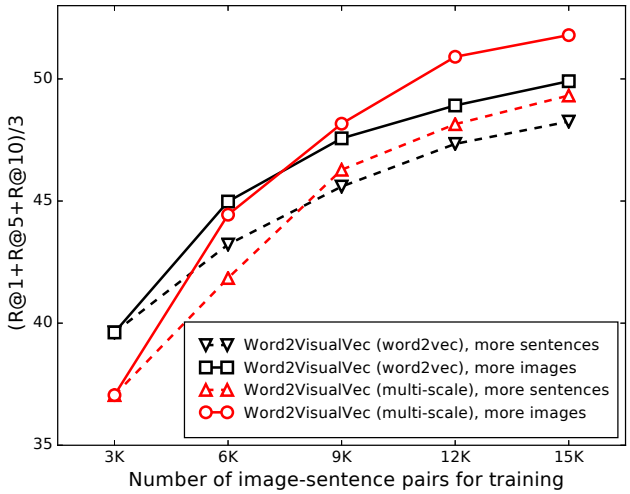

Fig. 3. Performance curves of two Word2VisualVec models on the Flickr30k test set, as the amount of image-sentence pairs for training increases. For both models, adding more training images gives better performance compared to adding more training sentences.

performance on the validation set. While its learning capacity increases as the model goes deeper, the chance of overfitting also increases. To improve generalization we also tried l_2 regularization on the network weights. This tactic brings a marginal improvement, yet introduces extra hyper parameters. So we did not go further in that direction. Overall the three-layer Word2VisualVec strikes the best balance between model capacity and generalization ability, so we use this network configuration in what follows.

How to vectorize an input sentence? Table II shows the performance of Word2VisualVec given different sentence vectorization strategies. On both Flickr8k and Flickr30k, multi-scale sentence vectorization outperforms its single-scale counterparts. Different phenomenons are observed on MSVD, where sentence vectorization by word2vec alone performs the best. Although MSVD has more visual / sentence pairs than Flickr8k, it has a much less number of 1,200 visual examples for training. The result suggests that given a fixed amount of training pairs, having more visual examples is better. To verify this conjecture, we take from the Flickr30k training set a random subset of 3k images with one sentence per image. We then incrementally increase the amount of image / sentence pairs for training, using the following two strategies. One is to increase the number of sentences per image from 1 to 2, 3, 4, and 5 with the number of images fixed, while the other is to let the amount of images increase to 6k, 9k, 12k and 15k with the number of sentences per image fixed to one. As the performance curves in Fig. 3 show, given the same amount of training pairs, adding more images results in better models. The result is also instructive for more effective acquisition of training data for image and video caption retrieval.

How fast? We implement Word2VisualVec using Keras with theano backend. The three-layer model with multi-scale sentence vectorization takes about 1.3 hours to learn from the 30k image-sentence pairs in Flickr8k on a GeoForce GTX 1070 GPU. Predicting visual features for a given sentence is swift, at an averaged speed of 20 milliseconds. Retrieving captions from a pool of 5k sentences takes 8 milliseconds per test image. Based on the above evaluations we recommend Word2VisualVec that predicts the ResNet-152 feature, either using multi-scale sentence vectorization given adequate training data (over 2k training images with five sentences per image) or using word2vec based sentence vectorization for learning from scarce resources. Code and models of our approach and experiments will be released.

B. *Word2VisualVec versus word2vec*

Although our model is meant for caption retrieval, it essentially generates a new representation of text. How meaningful is this new representation as compared to word2vec? To answer this question, we take all the 5K test sentences from Flickr30k, vectorizing them by word2vec and Word2VisualVec, respectively. For a fair comparison, we let Word2VisualVec use the same word2vec as its first layer. Fig. 4 presents t-SNE visualizations of sentence distributions in the word2vec and Word2VisualVec spaces, showing that sentences describing the same image stay more close while sentences from distinct images are more distant in the latter space. Recall that sentences associated with the same image are meant for describing the same visual content. Moreover, since they were independently written by distinct users, the wording may vary across the users, requiring a text representation to capture shared semantics among distinct words. Word2VisualVec better handles such variance in captions as illustrated in the first two examples in Fig. 4 (e).

The last example in Fig. 4 (e) shows failures of both word2vec and Word2VisualVec, where the two sentences (#5 and #6) are supposed to be close. The large difference between their subject (*teenagers* versus *people*) and object (*shirt* versus *paper*) makes it difficult for Word2VisualVec to predict similar visual features from the two sentences. Actually, we find in the Word2VisualVec space that the sentence nearest to #5 is "A woman is completing a picture of a young woman" (which resembles subjects, *i.e., teenager* versus *young woman* and action, *i.e., holding paper or easel*) and the one to #6 is "Kids scale a wall as two other people watch" (which depicts similar subjects, *i.e., two people* and objects, *i.e., concrete* versus *wall*). This example shows the existence of large divergence between manually written descriptions of the same visual content, and thus the challenging nature of the caption retrieval problem.

C. *Word2VisualVec for multimodal querying*

Fig. 5 presents an example of Word2VisualVec's learned representation and its ability for multimodal query composition. Given the query image, its composed queries are obtained by subtracting and/or adding the visual features of the query words, as predicted by Word2VisualVec. A deep dream visualization [67] is performed on an average (gray) image guided by each composed query. Consider the query in the second row for instance, where we instruct the search to replace bicycle with motorbike via a textual specification. The predicted visual feature of word *bicycle* is subtracted (effect

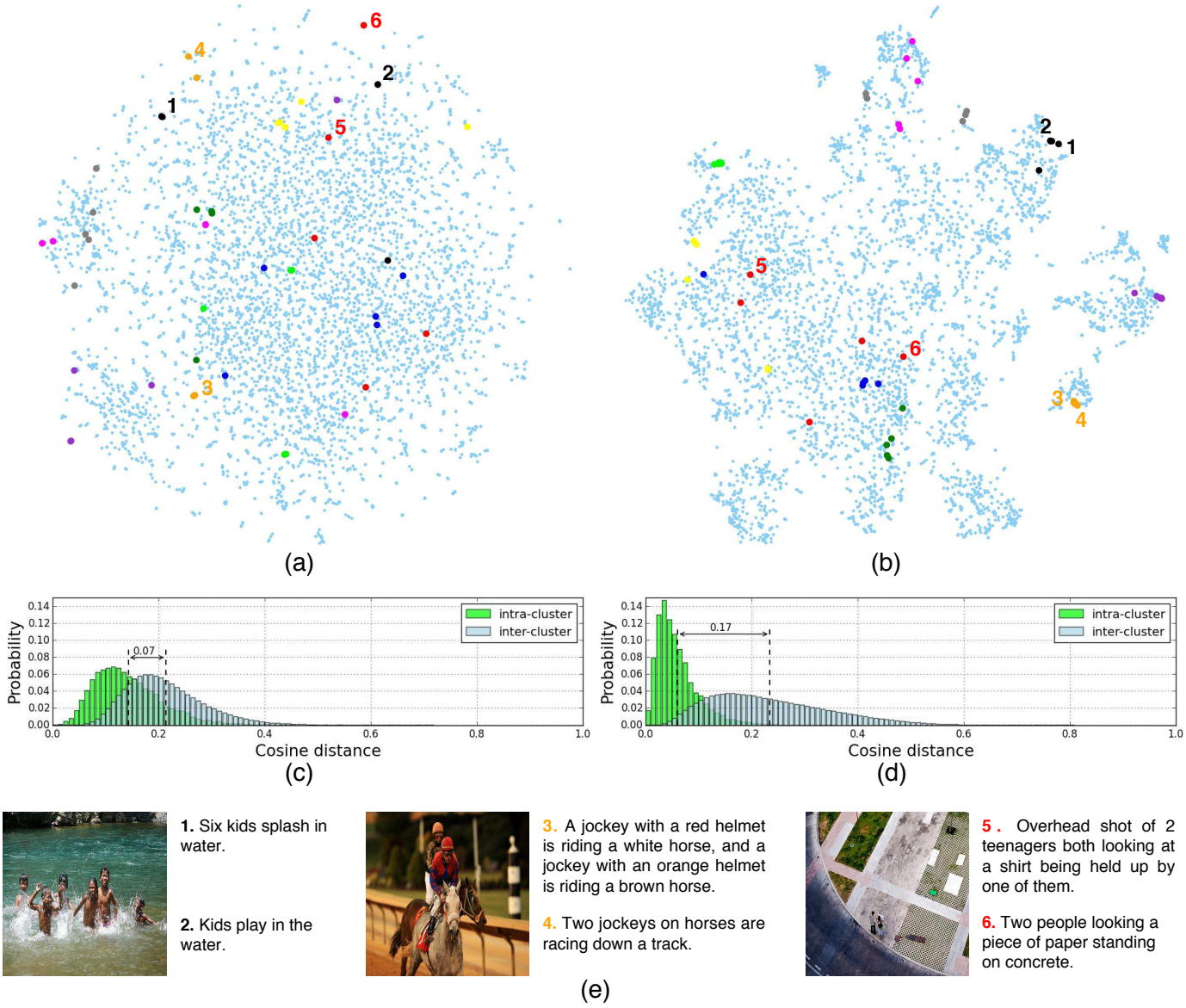

Fig. 4. **Word2VisualVec versus word2vec**. For the 5k test sentences from Flickr30k, we use t-SNE [66] to visualize their distribution in (a) the word2vec space and (b) the Word2VisualVec space obtained by mapping the word2vec vectors to the ResNet-152 features. Histograms of intra-cluster (*i.e.,* sentences describing the same image) and inter-cluster (*i.e.,* sentences from different images) distances in the two spaces are given in (c) and (d). Bigger colored dots indicate 50 sentences associated with 10 randomly chosen images, with exemplars detailed in (e). Together, the plots reveal that different sentences describing the same image stay closer, while sentences from different images are more distant in the Word2VisualVec space. Best viewed in color.

visible in first row) and the predicted visual feature of word *motorbike* is added. Imagery of motorbikes are indeed present in the dream. Hence, the nearest retrieved images emphasize on motorbikes in street scenes.

D. Comparison to the State-of-the-Art

Image caption retrieval. We compare a number of recently developed models for image caption retrieval [7], [8], [68], [44], [49], [9], [10]. Among them, [9], [10] have released their source code. So we re-train these two models with the same ResNet features we use. Table III presents the performance of the above models on both Flickr8k and Flickr30k. Word2VisualVec compares favorably against the state-of-the-art. Given the same visual feature, our model outperforms [9], [10], especially for $R@1$. Notice that Plummer *et al.* [68] employ extra bounding-box level annotations. Still our results are better. Indicating that we can expect further gains by including locality in the Word2VisualVec representation. As all the competitor models use joint subspaces, the results justify the viability of directly using the deep visual feature space for image caption retrieval.

Video caption retrieval. We also participated in the NIST TrecVid 2016 video caption retrieval task [43]. The test set consists of 1,915 videos collected from Twitter Vine. Each video is about 6 sec long. The videos were given to 8 annotators to generate a total of 3,830 sentences, with each

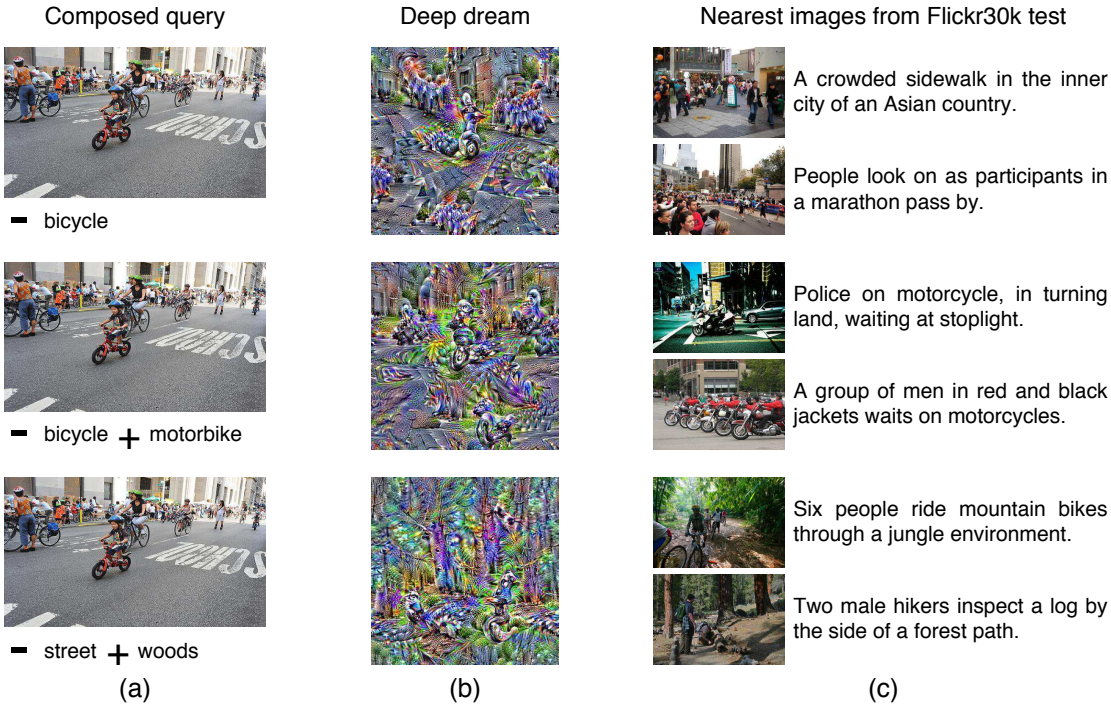

Fig. 5. **Word2VisualVec allows for multimodal query composition.** (a) For each multimodal query we visualize its predicted visual feature in (b) and show in (c) the nearest images and their sentences from the Flickr30k test set. Note the change in emphasis in (b), better viewed digitally in close-up.

TABLE III
STATE-OF-THE-ART FOR IMAGE CAPTION RETRIEVAL. ALL NUMBERS ARE FROM THE CITED PAPERS EXCEPT FOR [9], [10], BOTH RE-TRAINED USING THEIR CODE WITH THE SAME RESNET FEATURES WE USE. WORD2VISUALVEC OUTPERFORMS RECENT ALTERNATIVES.

	Flickr8k			Flickr30k		
	R@1	R@5	R@10	R@1	R@5	R@10
Ma *et al.* [7]	24.8	53.7	67.1	33.6	64.1	74.9
Kiros *et al.* [9]	23.7	53.1	67.3	32.9	65.6	77.1
Klein *et al.* [8]	31.0	59.3	73.7	35.0	62.0	73.8
Lev *et al.* [49]	31.6	61.2	74.3	35.6	62.5	74.2
Plummer *et al.* [68]	–	–	–	39.1	64.8	76.4
Wang *et al.* [44]	–	–	–	40.3	68.9	79.9
Vendrov *et al.* [10]	27.5	56.5	69.2	41.3	71.0	80.8
Word2VisualVec	**36.3**	**66.4**	**78.2**	**45.9**	**71.9**	**81.3**

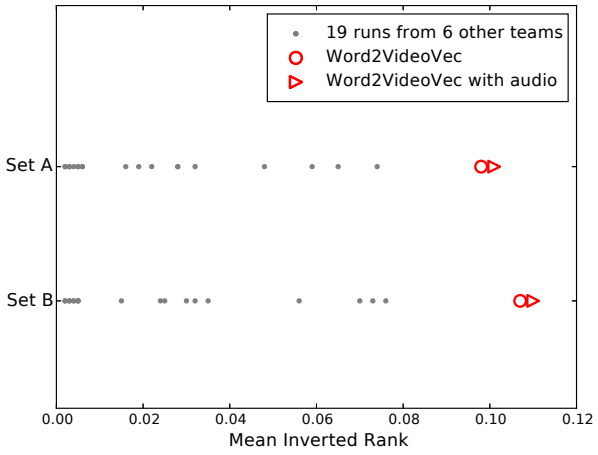

Fig. 6. **State-of-the-art for video caption retrieval** in the TrecVid 2016 benchmark, showing the good performance of Word2VideoVec compared to 19 alternative approaches, which can be further improved by predicting the visual-audio feature.

video associated with two sentences written by two different annotators. The sentences have been split into two equal-sized subsets, set A and set B, with the rule that sentences describing the same video are not in the same subset. Per test video, participants are asked to rank all sentences in the two subsets. Notice that we have no access to the ground-truth, as the test set is used for blind testing by the organizers only. NIST also provides a training set of 200 videos, which we consider insufficient for training Word2VideoVec. Instead, we learn the network parameters using video-text pairs from MSR-VTT [69], with hyper-parameters tuned on the provided TrecVid training set. By the time of TrecVid submission, we used GoogLeNet-shuffle as the visual feature, a 1,024-dim bag of MFCC as the audio feature, and word2vec for sentence vectorization. The performance metric is Mean Inverted Rank at which the annotated item is found. Higher mean inverted rank means better performance.

As shown in Fig. 6, with Mean Inverted Rank ranging from 0.097 to 0.110, Word2VideoVec leads the evaluation on both set A and set B in the context of 21 submissions from seven teams worldwide. Moreover, the results can be further improved by predicting the visual-audio feature. Some qualitative image and video caption retrieval results are shown in Fig. 7.

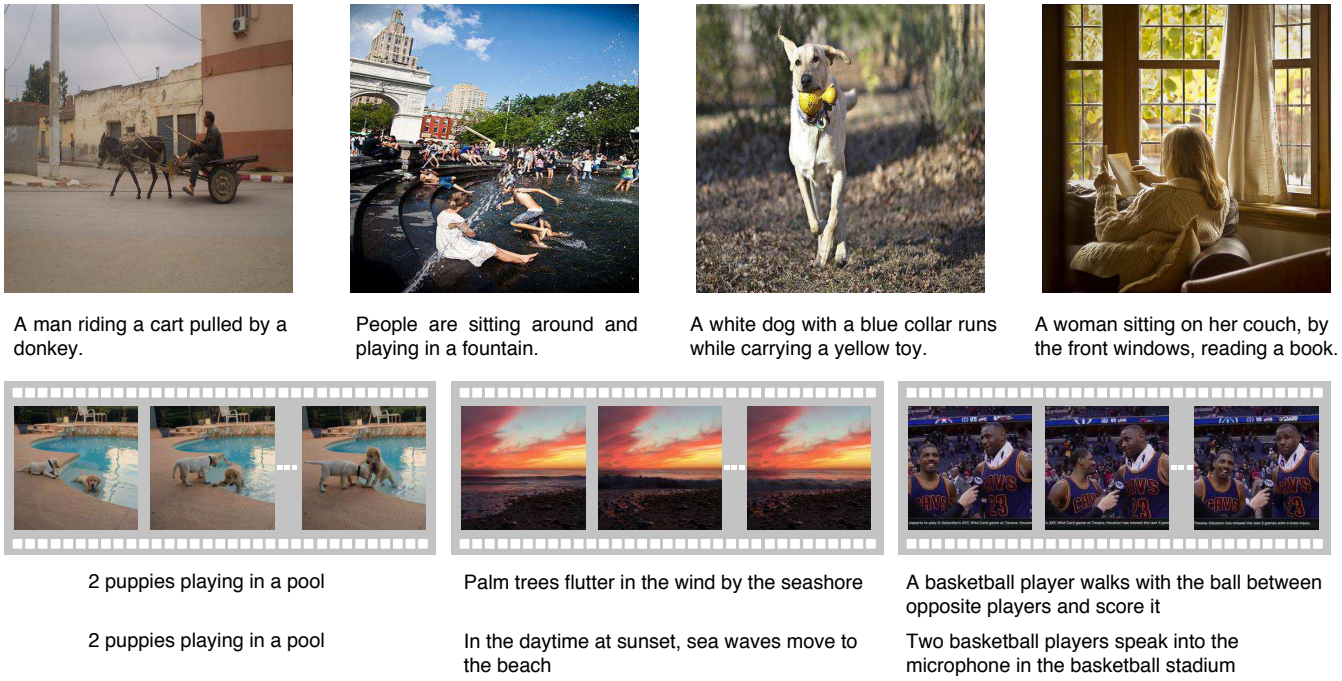

Fig. 7. Some image and video caption retrieval results by this work. The last row are the sentences retrieved by Word2VideoVec with audio, showing that adding audio sometimes help describe acoustics, e.g. *sea wave* and *speak*.

V. Conclusions

This paper shows the viability of resolving image and video caption retrieval in a visual feature space exclusively. We contribute Word2VisualVec, which is capable of transforming a natural language sentence to a meaningful visual feature representation. Compared to the word2vec space, sentences describing the same image tend to stay closer, while sentences from different images are more distant in the Word2VisualVec space. As the sentences are meant for describing visual content, the new textual encoding captures both semantic and visual similarities. Word2VisualVec also supports multimodal query composition, by subtracting and/or adding the predicted visual features of specific words to a given query image. What is more the Word2VisualVec is easily generalized to predict a visual-audio representation from text for video caption retrieval. For state-of-the-art results, while considering the availability of training data, we suggest to rely on either multi-scale sentence vectorization or word2vec based sentence vectorization, before predicting the ResNet feature with Word2VisualVec.

Acknowledgment

This work was supported by National Science Foundation of China (No. 61672523).

References

[1] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. G. Lanckriet, R. Levy, and N. Vasconcelos, "A new approach to cross-modal multimedia retrieval," in *MM*, 2010.
[2] F. Feng, X. Wang, and R. Li, "Cross-modal retrieval with correspondence autoencoder," in *MM*, 2014.
[3] L. Zhang, B. Ma, G. Li, Q. Huang, and Q. Tian, "Pl-ranking: a novel ranking method for cross-modal retrieval," in *MM*, 2016.
[4] Y. Yu, H. Ko, J. Choi, and G. Kim, "End-to-end concept word detection for video captioning, retrieval, and question answering," in *CVPR*, 2017.
[5] Y. He, S. Xiang, C. Kang, J. Wang, and C. Pan, "Cross-modal retrieval via deep and bidirectional representation learning," *TMM*, vol. 18, no. 7, pp. 1363–1377, 2016.
[6] Y. Gong, L. Wang, M. Hodosh, J. Hockenmaier, and S. Lazebnik, "Improving image-sentence embeddings using large weakly annotated photo collections," in *ECCV*, 2014.
[7] L. Ma, Z. Lu, L. Shang, and H. Li, "Multimodal convolutional neural networks for matching image and sentence," in *ICCV*, 2015.
[8] B. Klein, G. Lev, G. Sadeh, and L. Wolf, "Associating neural word embeddings with deep image representations using fisher vectors," in *CVPR*, 2015.
[9] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," *TACL*, 2015.
[10] I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun, "Order-embeddings of images and language," in *ICLR*, 2016.
[11] R. Xu, C. Xiong, W. Chen, and J. J. Corso, "Jointly modeling deep video and compositional text to bridge vision and language in a unified framework," in *AAAI*, 2015.
[12] V. Ordonez, G. Kulkarni, and T. L. Berg, "Im2text: Describing images using 1 million captioned photographs," in *NIPS*, 2011.
[13] J. Devlin, S. Gupta, R. Girshick, M. Mitchell, and C. L. Zitnick, "Exploring nearest neighbor approaches for image captioning," *arXiv preprint arXiv:1505.04467*, 2015.
[14] S. Yagcioglu, E. Erdem, A. Erdem, and R. Cakici, "A distributed representation based query expansion approach for image captioning," in *ACL*, 2015.
[15] E. Mansimov, E. Parisotto, J. L. Ba, and R. Salakhutdinov, "Generating images from captions with attention," in *ICLR*, 2016.
[16] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *ICML*, 2016.
[17] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification using deep convolutional neural networks," in *NIPS*, 2012.
[18] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *MM*, 2014.
[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.
[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[23] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *CVPR Workshop*, 2014.

[24] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in *ICLR*, 2014.

[25] G. Ye, Y. Li, H. Xu, D. Liu, and S.-F. Chang, "EventNet: a large scale structured concept library for complex event detection in video," in *MM*, 2015.

[26] Z. Wu, Y.-G. Jiang, X. Wang, H. Ye, and X. Xue, "Multi-stream multi-class fusion of deep networks for video classification," in *MM*, 2016.

[27] K. Cho, A. Courville, and Y. Bengio, "Describing multimedia content using attention-based encoder-decoder networks," *TMM*, vol. 17, no. 11, pp. 1875–1886, 2015.

[28] L. Jiang, S.-I. Yu, D. Meng, Y. Yang, T. Mitamura, and A. Hauptmann, "Fast and accurate content-based semantic search in 100m Internet videos," in *MM*, 2015.

[29] X. Jiang, F. Wu, X. Li, Z. Zhao, W. Lu, S. Tang, and Y. Zhuang, "Deep compositional cross-modal learning to rank via local-global alignment," in *MM*, 2015.

[30] X. Shang, H. Zhang, and T.-S. Chua, "Deep learning generic features for cross-media retrieval," in *MMM*, 2016.

[31] J. Chen and C.-W. Ngo, "Deep-based ingredient recognition for cooking recipe retrieval," in *MM*, 2016.

[32] Y. Hua, S. Wang, S. Liu, A. Cai, and Q. Huang, "Cross-modal correlation learning by adaptive hierarchical semantic aggregation," *TMM*, vol. 18, no. 6, pp. 1201–1216, 2016.

[33] A. Frome, G. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov, "DeViSE: A deep visual-semantic embedding model," in *NIPS*, 2013.

[34] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng, "Grounded compositional semantics for finding and describing images with sentences," *TACL*, vol. 2, pp. 207–218, 2014.

[35] L. Zhang, B. Ma, G. Li, Q. Huang, and Q. Tian, "Cross-modal retrieval using multiordered discriminative structured subspace learning," *TMM*, vol. 19, no. 6, pp. 1220–1233, 2017.

[36] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *ICLR*, 2013.

[37] M. Jain, J. C. van Gemert, T. Mensink, and C. G. M. Snoek, "Objects2action: Classifying and localizing actions without any video example," in *ICCV*, 2015.

[38] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *ICCV*, 2015.

[39] F. Eyben, F. Weninger, F. Gross, and B. Schuller, "Recent developments in openSMILE, the Munich open-source multimedia feature extractor," in *MM*, 2013.

[40] M. Hodosh, P. Young, and J. Hockenmaier, "Framing image description as a ranking task: Data, models and evaluation metrics," *JAIR*, vol. 47, no. 1, pp. 853–899, 2013.

[41] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *TACL*, vol. 2, pp. 67–78, 2014.

[42] D. L. Chen and W. B. Dolan, "Collecting highly parallel data for paraphrase evaluation," in *ACL*, 2011.

[43] G. Awad, J. Fiscus, D. Joy, M. Michel, A. Smeaton, W. Kraaij, G. Quenot, M. Eskevich, R. Aly, R. Ordelman, G. Jones, B. Huet, and M. Larson, "Trecvid 2016: Evaluating video search, video event detection, localization, and hyperlinking," in *TRECVID*, 2016.

[44] L. Wang, Y. Li, and S. Lazebnik, "Learning deep structure-preserving image-text embeddings," in *CVPR*, 2016.

[45] M. Otani, Y. Nakashima, E. Rahtu, J. Heikkilä, and N. Yokoya, "Learning joint representations of videos and sentences with web image search," in *ECCV Workshop*, 2016.

[46] T. Yao, T. Mei, and C.-W. Ngo, "Learning query and image similarities with ranking canonical correlation analysis," in *ICCV*, 2015.

[47] J. L. Ba, K. Swersky, S. Fidler, and R. Salakhutdinov, "Predicting deep zero-shot convolutional neural networks using textual descriptions," in *ICCV*, 2015.

[48] Q. You, L. Cao, H. Jin, and J. Luo, "Robust visual-textual sentiment analysis: When attention meets tree-structured recursive neural networks," in *MM*, 2016.

[49] G. Lev, G. Sadeh, B. Klein, and L. Wolf, "Rnn fisher vectors for action recognition and image annotation," in *ECCV*, 2016.

[50] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: a neural image caption generator," in *CVPR*, 2015.

[51] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille, "Deep captioning with multimodal recurrent neural networks (m-rnn)," in *ICLR*, 2015.

[52] C. Wang, H. Yang, C. Bartz, and C. Meinel, "Image captioning with deep bidirectional LSTMs," in *MM*, 2016.

[53] J. Dong, X. Li, W. Lan, Y. Huo, and C. G. M. Snoek, "Early embedding and late reranking for video captioning," in *MM*, 2016.

[54] X. Li, S. Liao, W. Lan, X. Du, and G. Yang, "Zero-shot image tagging by hierarchical semantic embedding," in *SIGIR*, 2015.

[55] S. Cappallo, T. Mensink, and C. G. M. Snoek, "Image2emoji: Zero-shot emoji prediction for visual media," in *MM*, 2015.

[56] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *EMNLP*, 2014.

[57] D. Grangier and S. Bengio, "A discriminative kernel-based approach to rank images from text queries," *TPAMI*, vol. 30, no. 8, pp. 1371–1384, 2008.

[58] B. Bai, J. Weston, D. Grangier, R. Collobert, K. Sadamasa, Y. Qi, C. Cortes, and M. Mohri, "Polynomial semantic indexing," in *NIPS*, 2009.

[59] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *CVPR*, 2015.

[60] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude." COURSERA: Neural Networks for Machine Learning, 2012.

[61] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, "Translating videos to natural language using deep recurrent neural networks," in *NAACL-HLT*, 2015.

[62] P. Mettes, D. C. Koelma, and C. G. M. Snoek, "The ImageNet shuffle: Reorganized pre-training for video event detection," in *ICMR*, 2016.

[63] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui, "Jointly modeling embedding and translation to bridge video and language," in *CVPR*, 2016.

[64] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu, "Video paragraph captioning using hierarchical recurrent neural networks," in *CVPR*, 2016.

[65] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *CVPR*, 2014.

[66] L. van de Maaten and G. Hinton, "Visualizing data using t-sne," *JMLR*, vol. 9, pp. 2579–2605, 2008.

[67] A. Mordvintsev, C. Olah, and M. Tyka, "Inceptionism: Going deeper into neural networks," Google Research Blog, 2015.

[68] B. Plummer, L. Wang, C. Cervantes, J. Caicedo, J. Hockenmaier, and S. Lazebnik, "Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models," in *ICCV*, 2015.

[69] J. Xu, T. Mei, T. Yao, and Y. Rui, "MSR-VTT: A large video description dataset for bridging video and language," in *CVPR*, 2016.