

Learning Joint Representations of Videos and Sentences with Web Image Search

Mayu Otani¹, Yuta Nakashima¹, Esa Rahtu²,
Janne Heikkilä², and Naokazu Yokoya¹

¹ Graduate School of Information Science, Nara Institute of Science and Technology
`{otani.mayu.ob9,n-yuta,yokoya}@is.naist.jp`

² Center for Machine Vision and Signal Analysis, University of Oulu
`{erahtu,jth}@ee.oulu.fi`

Abstract. Our objective is video retrieval based on natural language queries. In addition, we consider the analogous problem of retrieving sentences or generating descriptions given an input video. Recent work has addressed the problem by embedding visual and textual inputs into a common space where semantic similarities correlate to distances. We also adopt the embedding approach, and make the following contributions: First, we utilize web image search in sentence embedding process to disambiguate fine-grained visual concepts. Second, we propose embedding models for sentence, image, and video inputs whose parameters are learned simultaneously. Finally, we show how the proposed model can be applied to description generation. Overall, we observe a clear improvement over the state-of-the-art methods in the video and sentence retrieval tasks. In description generation, the performance level is comparable to the current state-of-the-art, although our embeddings were trained for the retrieval tasks.

Keywords: Video retrieval, sentence retrieval, multimodal embedding, neural network, image search, representation learning

1 Introduction

During the last decade, the Internet has become an increasingly important distribution channel for videos. Video hosting services like YouTube, Flickr, and Vimeo have millions of users uploading and watching content every day. At the same time, powerful search methods have become essential to make good use of such vast databases. By analogy, without textual search tools like Google or Bing, it would be nearly hopeless to find information from the websites.

Our objective is to study the problem of retrieving video clips from a database using natural language queries. In addition, we consider the analogous problem of retrieving sentences or generating descriptions based on a given video clip. We are particularly interested in learning appropriate representations for both visual and textual inputs. Moreover, we intend to leverage the supporting information provided by the current image search approaches.

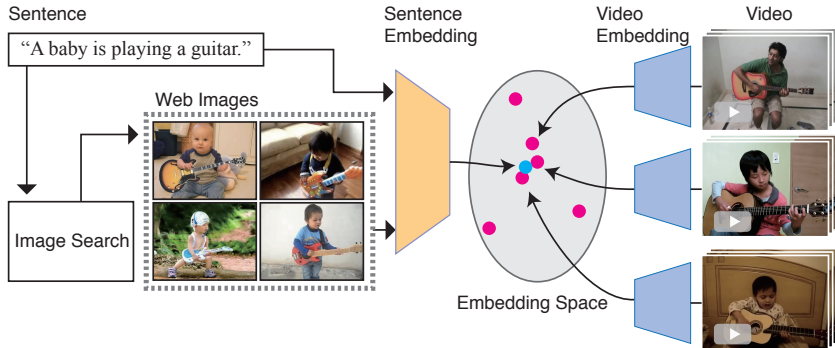


Fig. 1. An overview of our approach. Left side illustrates the image search results for a query “A baby is playing a guitar”. Images highlight evidence of objects (“baby”, “guitar”) and actions (“playing”). Right side shows the most relevant videos in the YouTube dataset [1] obtained by ranking the clips according to Euclidean distance to the query sentence in an embedding space.

This topic has recently received plenty of attention in the community, and papers have presented various approaches to associate visual and textual data. One direction to address this problem is to utilize metadata that can be directly compared with queries. For instance, many web image search engines evaluate the relevance of an image based on the similarity of the query sentence with the user tags or the surrounding HTML text [4]. For sentence retrieval, Ordonez *et al.* [21] proposed to compare an image query and visual metadata with sentences.

While these methods using comparable metadata have demonstrated impressive results, they do not perform well in cases where appropriate metadata is limited or not available. Moreover, they rely strongly on the assumption that the associated visual and textual data in the database is relevant to each other. These problems are more apparent in the video retrieval task since video distribution portals like YouTube often provide less textual descriptions compared to other web pages. Furthermore, available descriptions (e.g. title) often cover only a small portion of the entire visual content in a video.

An alternative approach would be to compare textual and visual inputs directly. In many approaches, this is enabled by embedding the corresponding representations into a common vector space in such a way that the semantic similarity of the original inputs would be directly reflected in their distance in the embedding space (Fig. 1). Recent work [27, 16] has proposed deep neural network models for performing such embeddings. The results are promising, but developing powerful joint representations still remains a challenge.

In this paper, we propose a new embedding approach for sentence and video inputs that combines the advantages of the metadata-based web image search and deep neural network-based representation learning. More precisely, we use a standard search engine to obtain a set of supplementary images for each query sentence. Then, we pass the sentence and the retrieved images to a two-branch

neural network that produces the sentence embedding. The video inputs are embedded into the same space using another neural network. The network parameters are trained jointly so that videos and sentences with similar semantic content are mapped to close points. Figure 1 illustrates the overall architecture of our approach. The experiments indicate a clear improvement over the current state-of-the-art baseline methods.

Our main contributions are as follows:

- We present an embedding approach for video retrieval that incorporates web image search results to disambiguate fine-grained visual concepts in query sentences.
- We introduce neural network-based embedding models for video, sentence, and image inputs whose parameters can be learned jointly. Unlike previous work that uses only videos and sentences, we utilize a sentence and corresponding web images to compute the sentence embedding.
- We demonstrate a clear improvement over the state-of-the-art in the video and sentence retrieval tasks with the YouTube dataset [1].
- We demonstrate description generation as an example of possible applications of our video embeddings. We observed that the performance is comparable with the state-of-the-art. This indicates that video contents are efficiently encoded into our video embeddings.

2 Related Work

Visual and Language Retrieval: Due to the explosive growth of images and videos on the web, visual retrieval has become a hot topic in computer vision and machine learning [4, 20]. Several recent approaches for joint representation learning enable direct comparison among different multimodalities. Farhadi *et al.* [7] introduced triplets of labels on object, action, and scene as joint representations for images and sentences. Socher *et al.* [27] proposed to embed representations of images and labels into a common embedding space. For videos, the approach proposed by Lin *et al.* [18] associates a parsed semantic graph of a query sentence and visual cues based on object detection and tracking.

The recent success of deep convolutional neural networks (CNNs) together with large-scale visual datasets [22, 2, 25] has resulted in several powerful representation models for images [5, 33, 35]. These CNN-based methods have been successfully applied to various types of computer vision tasks, such as object detection [10, 23], video summarization [12], and image description generation [32, 6].

Deep neural networks have also been used in the field of natural language processing [17, 16]. For example, Kiros *et al.* [16] proposed sentence representation learning based on recurrent neural networks (RNNs). They also demonstrated image and sentence retrieval by matching sentence and image representations with jointly learned linear transformations.

Representation learning using deep neural networks is explored in many tasks [3, 19, 9, 14, 34, 37]. Frome *et al.* [9] proposed image classification by computing

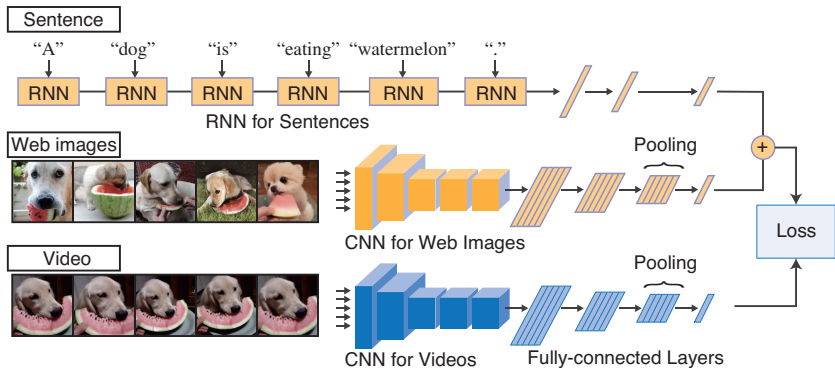


Fig. 2. Illustration of our video and sentence embedding. The orange component is the sentence embedding model that takes a sentence and corresponding web images as input. Video embedding model is denoted by the blue component.

similarity between joint representations of images and labels, and Zhu *et al.* [37] addressed alignment of a movie and sentences in a book using joint representations for video clips and sentences. Their approach also computes similarity between sentences and subtitles of video clips to improve the alignment of video clips and sentences.

Our approach is the closest to work by Xu *et al.* [34]. They represent a sentence by a subject, verb, and object (SVO) triplet, and embed sentences as well as videos to a common vector space using deep neural networks. The main difference between ours and the work [34] is the use of an RNN to encode a sentence and supplementary web images. The use of an RNN enables our model to encode all words in a sentence and capture details of the sentence, such as an object’s attributes and scenes, together with corresponding web images.

Exploiting Image Search: The idea of exploiting web image search is adopted in many tasks, including object classification [8] and video summarization [28]. These approaches collect a vast amount of images from the web and utilize them to extract canonical visual concepts. Recent label prediction for images by Johnson *et al.* [13] infers tags of target images by mining relevant Flickr images based on their metadata, such as user tags and photo groups curated by users. The relevant images serve as priors on tags for the target image. A similar motivation drives us to utilize web images for each sentence, which can disambiguate visual concepts of the sentence and highlight relevant target videos.

3 Proposed Approach

We propose neural network-based embedding models for the video and sentence retrieval tasks. In order to enhance the sentence embedding, we retrieve relevant

web images that are assumed to disambiguate semantics of the sentence. For example, the word “keyboard” can be interpreted as a musical instrument or an input device for computers. If the word comes with “play,” the meaning of “keyboard” narrows down to a musical instrument. This means that a specific combination of words can reduce the possible visual concepts relevant to the sentence, which may not be fully encoded even with the state-of-the-art RNN-based approach like [16].

We propose to take this into account by using web image search results. Since most image search engines use surrounding text to retrieve images, we can expect that they are responsive to such word combinations. Consequently, we retrieve web images using the input sentence as a query and download the results. The web images are fused with the input sentence by applying a two-branch neural network as shown in Fig. 2. Videos are also encoded by applying a neural network-based video embedding model. Relevance between sentence and video inputs is directly computed in the common embedding space using Euclidean distances. We jointly train our embedding models using video-sentence pairs by minimizing the contrastive loss [3].

3.1 Video Embedding

We extract frames from a video at 1 fps as in [34] and feed them to a CNN-based video embedding model. In our approach, we employ two CNN architectures: 19-layer VGG [26] and GoogLeNet [29], both of which are pre-trained on ImageNet [25]. We replace the classifier layer in each model with two fully-connected layers. Specifically, we compute activations of the VGG’s fc7 layer or the GoogLeNet’s inception 5b layer and feed them to additional embedding layers.

Let $X = \{x_i \mid i = 1, \dots, M\}$ be a set of frames x_i , and $\text{CNN}(x_i) \in \mathbb{R}^{d_v}$ be an activation of a CNN ($d_v=4,096$ for VGG, and $d_v=1,024$ for GoogLeNet). The video embedding $\phi_v(X) \in \mathbb{R}^{d_e}$ is computed by:

$$\phi_v(X) = \frac{1}{M} \sum_{x_i \in X} \tanh(W_{v_2} \tanh(W_{v_1} \text{CNN}(x_i) + b_{v_1}) + b_{v_2}). \quad (1)$$

Here, $W_{v_1} \in \mathbb{R}^{d_h \times d_v}$, $b_{v_1} \in \mathbb{R}^{d_h}$, $W_{v_2} \in \mathbb{R}^{d_e \times d_h}$, and $b_{v_2} \in \mathbb{R}^{d_e}$ are the learnable parameters of the fully-connected layers.

3.2 Sentence and Web Image Embedding

The sentence embedding model consists of two branches that merge the outputs of a CNN-based network for web images and an RNN-based network for a sentence. Before computing the sentence embedding, we download top- K results of web image search with the input sentence as a query. Let $Z = \{z_j \mid j = 1, \dots, K\}$ be a set of web images. We utilize the same architecture as the video embedding and compute an intermediate representation $e_z \in \mathbb{R}^{d_e}$ that integrates the web images as:

$$e_z = \frac{1}{K} \sum_{z_j \in Z} \tanh(W_{z_2} \tanh(W_{z_1} \text{CNN}(z_j) + b_{z_1}) + b_{z_2}), \quad (2)$$

where $W_{z_1} \in \mathbb{R}^{d_h \times d_v}$, $b_{z_1} \in \mathbb{R}^{d_h}$, $W_{z_2} \in \mathbb{R}^{d_e \times d_h}$, and $b_{z_2} \in \mathbb{R}^{d_e}$ are the learnable parameters of the two fully-connected layers.

We encode sentences into vector representations using skip-thought that is an RNN pre-trained with a large-scale book corpus [16]. Let $Y = \{y_t \mid t = 1, \dots, T_Y\}$ be the input sentence, where y_t is the t -th word in the sentence, and T_Y is the number of words in the sentence Y . Skip-thought takes a sequence of word vectors $w_t \in \mathbb{R}^{d_w}$ computed from a word input y_t as in [16] and produces hidden state $h_t \in \mathbb{R}^{d_s}$ at each time step t as:

$$r_t = \sigma(W_r w_t + U_r w_{t-1}), \quad (3)$$

$$i_t = \sigma(W_i w_t + U_i h_{t-1}), \quad (4)$$

$$a_t = \tanh(W_a w_t + U_a (r_t \odot h_{t-1})), \quad (5)$$

$$h_t = (1 - i_t) \odot h_{t-1} + i_t \odot a_t, \quad (6)$$

where σ is the sigmoid activation function, and \odot is the component-wise product. The parameters W_r, W_i, W_a, U_r, U_i , and U_a are $d_s \times d_w$ matrices. Sentence Y is encoded into the hidden state after processing the last word w_{T_Y} , *i.e.*, h_{T_Y} . We use combine-skip in [16], which is a concatenation of outputs from two separate RNNs trained with different datasets. We denote the output of combine-skip from sentence Y by $s_Y \in \mathbb{R}^{d_c}$, where $d_c=4,800$.

We also compute an intermediate representation e_s for sentence Y as:

$$e_s = \tanh(W_{s_2} \tanh(W_{s_1} s_Y + b_{s_1}) + b_{s_2}), \quad (7)$$

where $W_{s_1} \in \mathbb{R}^{d_h \times d_c}$, $b_{s_1} \in \mathbb{R}^{d_h}$, $W_{s_2} \in \mathbb{R}^{d_e \times d_h}$, and $b_{s_2} \in \mathbb{R}^{d_e}$ are the learnable parameters of sentence embedding.

Once the outputs e_s and e_z of each branch in our sentence embedding model are computed, they are merged into a sentence embedding $\phi_s(Y, Z)$ as:

$$\phi_s(Y, Z) = \frac{1}{2}(e_s + e_z). \quad (8)$$

By this simple mixture of e_s and e_z , the sentence and web images directly influence the sentence embedding.

3.3 Joint Learning of Embedding Models

We jointly train both embedding ϕ_v and ϕ_s using pairs of videos and associated sentences in a training set by minimizing the contrastive loss function [3]. In our approach, the contrastive loss decreases when embeddings of videos and sentences with similar semantics get closer to each other in the embedding space, and those with dissimilar semantics get farther apart.

The training process requires a set of positive and negative video-sentence pairs. A positive pair contains a video and a sentence that are semantically relevant, and a negative pair contains irrelevant ones. Let $\{(X_n, Y_n) \mid n = 1, \dots, N\}$ be the set of positive pairs. Given a positive pair (X_n, Y_n) , we sample irrelevant sentences $\mathcal{Y}'_n = \{Y'_f \mid f = 1, \dots, N_c\}$ and videos $\mathcal{X}'_n = \{X'_g \mid g = 1, \dots, N_c\}$ from

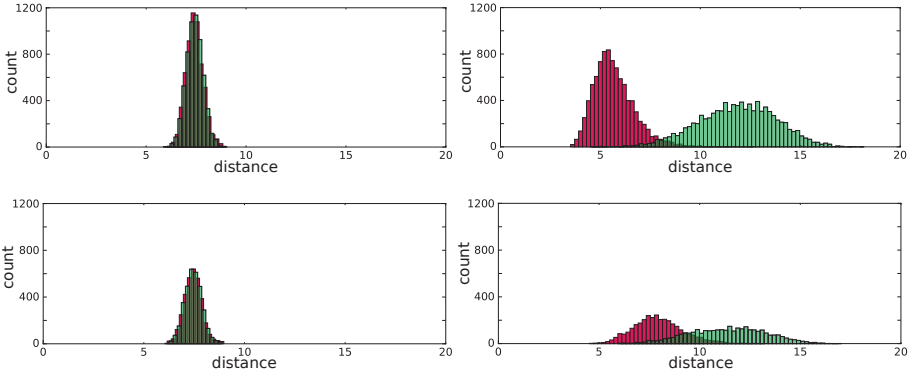


Fig. 3. Histograms of pairwise distances before training (left) and after training (right). Top row: Histograms of the training set. Bottom row: Histograms of the test set. Red represents positive pairs, and green represents negative pairs.

the training set, which are used to build two sets of negative pairs $\{(X_n, Y'_f) \mid Y'_f \in \mathcal{Y}'_n\}$ and $\{(X'_g, Y_n) \mid X'_g \in \mathcal{X}'_n\}$. In our approach, we set the size of negative pairs N_c to 50. We train the parameters of embedding ϕ_v and ϕ_s by minimizing the contrastive loss defined as:

$$\begin{aligned}
 Loss(X_n, Y_n) = \frac{1}{1+2N_c} & \left\{ d(X_n, Y_n) \right. \\
 & + \sum_{Y'_f \in \mathcal{Y}'_n} \max(0, \alpha - d(X_n, Y'_f)) \\
 & \left. + \sum_{X'_g \in \mathcal{X}'_n} \max(0, \alpha - d(X'_g, Y_n)) \right\}, \quad (9)
 \end{aligned}$$

$$d(X_i, Y_j) = \|\phi_v(X_i) - \phi_s(Y_j, Z_j)\|_2^2, \quad (10)$$

where Z_n is the web images corresponding to sentence Y_n . The hyperparameter α is a margin. Negative pairs with smaller distances than α are penalized. Margin α is set to the largest distance of positive pairs before training so that most negative pairs influence the model parameters at the beginning of training.

Figure 3 shows the histograms of distances of positive and negative pairs before and after training. The initial distance distributions of positive and negative pairs overlap. After training, the distributions are pulled apart. This indicates that the training process encourages videos and sentences in positive pairs to be mapped to closer points and those in negative ones to farther points.

The examples of positive and negative pairs in our test set with corresponding distances are shown in Fig. 4. The positive pair (a) and (b) are easy cases, in which sentences explicitly describe the video contents. The pair (c) is an example of hard cases. The sentence includes “a man” and “phone”, but the video actually shows two men, and a phone is occluded by a hand.

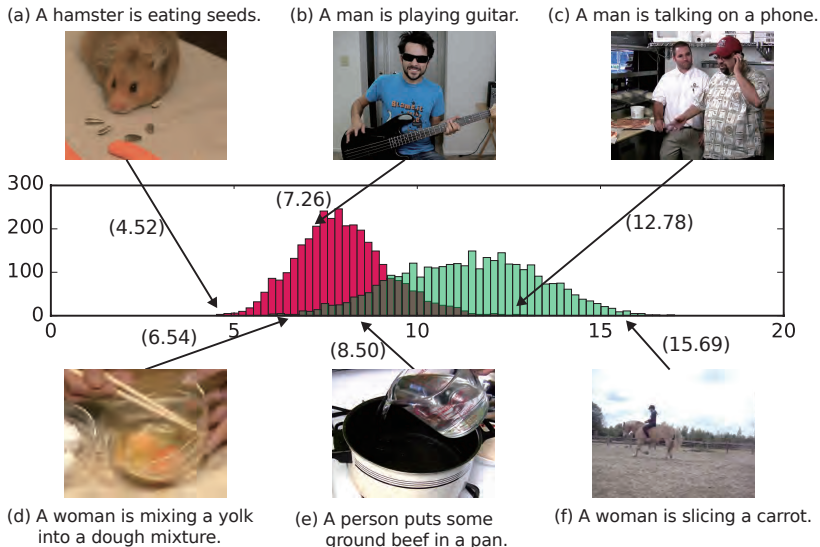


Fig. 4. Examples of positive (a)–(c) and negative (d)–(f) pairs in the test set with corresponding distances. The values (\cdot) are distances of the pairs. The plot is the histograms of distances of positive (red) and negative (green) pairs.

The pairs (d) and (e) are hard negative cases. The pair (d) shows partial matches of contents, such as the action “mixing” and the object “yolk.” Another negative pair (e) has a video and a sentence about cooking, although there is disagreement about details. As shown in these examples, the closer a video and a sentence are located in the embedding space, the more relevant they are. More examples can be found in the supplementary material.

4 Retrieval Experiments

4.1 Implementation Detail

With 19-layer VGG, the hidden layer size d_h of embedding ϕ_v and ϕ_s was set to 1,000 and the dimension of the embedding space d_e was set to 300. For model using GoogLeNet, we used $d_h = 600$ and $d_e = 300$.

We implemented our model using Chainer [30]. We used Adam [15] for optimization with a learning rate of 1×10^{-4} . The parameters of the CNNs and skip-thought were fixed. We applied dropout with a ratio of 0.5 to the input of the first and second layers of ϕ_v and ϕ_s . Our models were trained for 15 epochs, and their parameters were saved at every 100 updates. We took the model parameters whose performance was the best on the validation set.

4.2 Experimental Setup

Dataset: We used the YouTube dataset [1] consisting of 80K English descriptions for 1,970 videos. We first divided the dataset into 1,200, 100, and 670 videos for training, validation, and test, respectively, as in [35, 34, 11]. Then, we extracted five-second clips from each original video in a sliding-window manner. As a result, we obtained 8,001, 628, and 4,499 clips for the training, validation, and test sets, respectively. For each clip, we picked five ground truth descriptions out of those associated with its original video.

We collected top-5 image search results for each sentence using the Bing image search engine. We used a sentence modified by lowercasing and punctuation removal as a query. In order to eliminate cartoons and clip art, the image type was limited to photos using Bing API.

Video Retrieval: Given a video and a query sentence, we extracted five-second video clips from the video and computed Euclidean distances from the query to the clips. We used their median as the distance of the original video and the query. We ranked the videos based on the distance to each query and recorded the rank of the ground truth video. Since the test set has 670 videos, the probability of bringing the ground truth video at top-1 by random ranking is about 0.14%.

Sentence Retrieval: For the sentence retrieval task, we ranked sentences for each query video. We computed the distances between a sentence and a query video in the same way as the video retrieval task. Note that each video has five ground truth sentences; thus, we recorded the highest rank among them. The test set has 3,500 sentences.

Evaluation Metrics: We report recall rates at top-1, -5, and -10, the average and median rank, which are standard metrics employed in the retrieval evaluation. We found that some videos in the dataset had sentences whose semantics were almost the same (*e.g.*, “A group of women is dancing” and “Women are dancing”). For the video that is annotated with one of such sentences, the other sentence is treated as incorrect with the recall rates, which does not agree with human judges. Therefore, we employed additional evaluation metrics widely used in the description generation task, *i.e.*, CIDEr, BLUE@4, and METEOR [2]. They compute agreement scores in different ways using a retrieved sentence and a set of ground truth ones associated with a query video. Thus, these metrics give high scores for semantically relevant sentences even if they are not annotated to a query video. We computed the scores of the top ranked sentence for each video using the evaluation script provided in the Microsoft COCO Evaluation Server [2]. In our experiments, all ground truth descriptions for each original video are used to compute these scores.

Table 1. Video and sentence retrieval results. R@K is recall at top K results (higher values are better). aR and mR are the average and median of rank (lower values are better). Bold values denotes best scores of each metric.

Models	Video retrieval					Sentence retrieval				
	R@1	R@5	R@10	aR	mR	R@1	R@5	R@10	aR	mR
Random Ranking	0.14	0.79	1.48	335.92	333	0.22	0.69	1.32	561.32	439
VGG+VS	6.12	21.88	33.22	58.98	24	7.01	18.66	27.16	131.33	35
VGG+VI	4.03	13.70	21.40	94.62	48	5.67	17.91	28.21	116.86	38
VGG+ALL ₁	6.48	20.15	30.51	59.53	26	10.60	25.22	36.42	85.90	21
VGG+ALL ₂	5.97	21.31	32.54	56.01	24	8.66	22.84	33.13	100.14	29
GoogLeNet+VS	7.49	22.84	33.10	54.14	22	8.51	21.34	30.45	114.66	33
GoogLeNet+VI	4.24	16.42	24.96	84.48	41	6.87	17.31	30.00	96.78	30
GoogLeNet+ALL ₁	5.52	18.93	28.90	60.38	28	9.85	27.01	38.36	75.23	19
GoogLeNet+ALL ₂	7.67	23.40	34.99	49.08	21	9.85	24.18	33.73	85.16	22
ST [16]	2.63	11.55	19.34	106.00	51	2.99	10.90	17.46	241.00	77
DVCT [34]	-	-	-	224.10	-	-	-	-	236.27	-

Table 2. Evaluated scores of retrieved sentences. All values are reported in percentage (%). Higher scores are better.

Models	CIDEr	BLEU	METEOR
VGG+VS	30.44	27.16	25.74
VGG+VI	29.00	22.42	22.99
VGG+ALL ₁	42.52	30.81	27.77
VGG+ALL ₂	32.56	27.39	26.58
GoogLeNet+VS	33.82	26.97	25.99
GoogLeNet+VI	35.08	24.56	24.16
GoogLeNet+ALL ₁	43.52	29.99	27.48
GoogLeNet+ALL ₂	38.08	29.28	26.50

4.3 Effects of Each Component of Our Approach

In order to investigate the influence of each component of our approach, we tested some variations of our full model. The scores of the models on the video and sentence retrieval tasks are shown in Table 1. Our full model is denoted by ALL₂. ALL₁ is a variation of ALL₂ that computes embeddings with one fully-connected layer with the unit size of d_e . Comparison between ALL₁ and ALL₂ indicates that the number of fully-connected layers in embedding is not essential.

In order to evaluate the contributions of web images, we trained a model that does not use web images, *i.e.*, an embedding of a sentence Y is computed by $\phi_s(Y) = e_s$. We denote this model by VS. VGG+ALL₂ had better average rank than VGG+VS, and comparison between GoogLeNet+ALL₂ and GoogLeNet+VS also shows a clear advantage of incorporating web images.

We also tested a model without sentences, which is denoted by VI. It computes an embedding of web images by $\phi_s(Z) = e_z$. We investigated the effect of using both sentences and web images by comparing VI to our full model ALL₂.

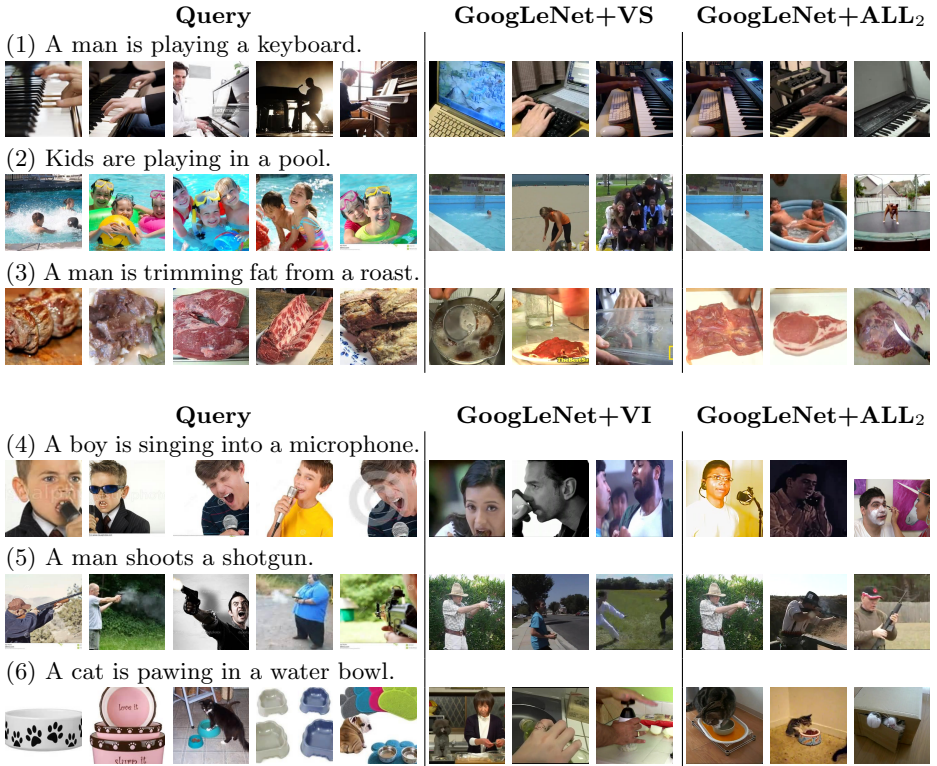


Fig. 5. Examples of video retrieval results. Left: Query sentence and web images. Center: Top-3 retrieved videos by GoogLeNet+VS and VI. Right: Top-3 retrieved videos by GoogLeNet+ALL₂.

The results show that sentences are necessary. The comparison between VI and VS also indicates that sentences provide main cues for the retrieval task.

The scores of retrieved sentences computed by CIDEr, BLEU@4, and METEOR are shown in Table 2. In all metrics, our model using both sentences and web images (ALL₁ and ALL₂) outperformed to other models (VS and VI). In summary, contributions by sentences and web images were non-trivial, and the best performance was achieved by using both of them.

Some examples of retrieved videos by GoogLeNet+VS, GoogLeNet+VI, and GoogLeNet+ALL₂ are shown in Fig. 5. These results suggest that web images reduced the ambiguity of queries’ semantics by providing hints on their visual concepts. For example, with sentence (1) “A man is playing a keyboard,” retrieval results of GoogleNet+VS includes two videos of a keyboard on a laptop as well as one on a musical instrument. On the other hand, all top-3 results by GoogleNet+ALL₂ are about musical instruments. Compared to GoogLeNet+VI, our full model obtained more videos with relevant content. Moreover, the result

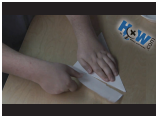



Query Video	GoogLeNet+Allz	GoogLeNet+VS
	<ol style="list-style-type: none"> 1. A man is cutting a paper. 2. A man is cutting a paper by hands. 3. Someone is cutting the carrot into small pieces. 	<ol style="list-style-type: none"> 1. Someone is cutting the carrot into small pieces. 2. A person cuts a sock with scissors. 3. An oriental lady is cutting a carrot into thin pieces.
	<ol style="list-style-type: none"> 1. A woman is talking while applying eyeshadow. 2. A woman applies Joker makeup to a man's face. 3. A woman is applying cosmetics to a man. 	<ol style="list-style-type: none"> 1. A woman is singing. 2. A woman is singing. 3. A woman wearing a headset is singing into a large microphone.
	<ol style="list-style-type: none"> 1. A pair of zebras are playing with each other. 2. The zebras are playing. 3. A pair of zebras is nuzzling. 	<ol style="list-style-type: none"> 1. Leopards are congregating. 2. A group of deers are crossing road. 3. A pair of zebras is nuzzling.
	<ol style="list-style-type: none"> 1. A man is playing keyboards. 2. A boy is playing a grand piano. 3. A boy is playing guitar. 	<ol style="list-style-type: none"> 1. A little boy is playing piano. 2. A little boy is playing a grand piano. 3. A boy is playing a piano.

Fig. 6. Examples of top-3 retrieved sentences. Left: Query videos. Center: Top-3 retrieved sentences by GoogLeNet+ALL₂. Right: Top-3 retrieved sentences by GoogLeNet+VS.

of query (6) indicates that our model can recover from irrelevant image search results by combining a query sentence.

Some examples of sentence retrieval results are shown in Fig. 6. While our full model may retrieve sentences that disagree with query videos in details, most of the retrieved sentences are relevant to query videos.

4.4 Comparison to Prior Work

The approach for image and sentence retrieval by Kiros *et al.* [16] applies linear transformations to CNN-based image and RNN-based sentence representations to embed them into a common space. Note that their model was designed for the image and sentence retrieval tasks; thus, we extracted the middle frame as a keyframe and trained the model with pairs of a keyframe and a sentence. Xu *et al.* [34] introduced neural network-based embedding models for videos and sentences. Their approach embeds videos and SVO triplets extracted from sentences into an embedding space. Kiros *et al.*'s and Xu *et al.*'s approaches are denoted by ST and DVCT, respectively.

Scores in Table 1 indicates that our model clearly outperformed prior work in both video and sentence retrieval tasks. There is a significant difference in performance of DVCT and others. ST and ours encode all words in a sentence, while DVCT only encodes its SVO triplets. This suggests that using all words in a sentence together with an RNN is necessary to get good embeddings.

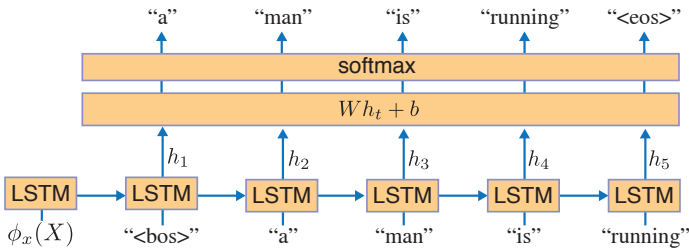


Fig. 7. Illustration of the decoder model. “<bos>” is a tag denoting the beginning of a sentence, and “<eos>” is the end of a sentence.

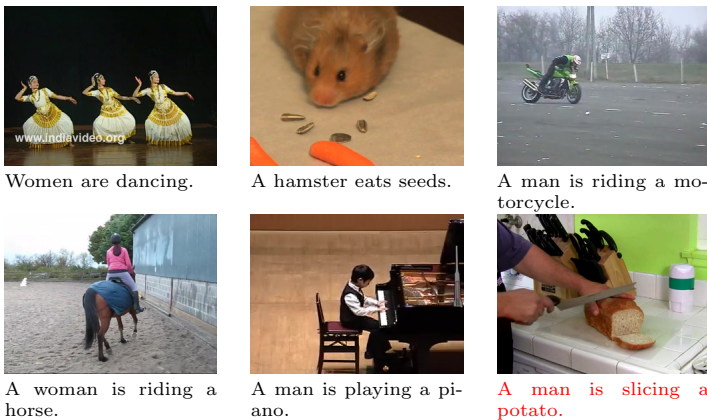


Fig. 8. Sentences generated from our video embeddings. The sentence in red is a failure.

Table 3. Scores of generated sentences. TVNL+Extra Data is the TVNL model pre-trained on the Flickr30k [36] and the COCO2014 [2] datasets.

Models	CIDEr	BLEU	METEOR
TVNL [31]	-	31.19	26.87
TVNL+Extra Data	-	33.29	29.07
DVETS [35]	51.67	41.92	29.60
Ours	41.62	33.69	28.47

5 Video Description Generation

Automatic description generation for images [32, 6] and videos [24, 31, 35] is another task to associate images or videos with sentences. As an application of our models, we performed the description generation task using our video embeddings. To analyze the information encoded by our video embedding, we trained a decoder that produces descriptions from our video embeddings. A basic approach for description generation is to use long-short term memory (LSTM) that

produces a sequence of probabilities over a vocabulary conditioned on visual representations [32, 31]. We trained an LSTM as a decoder of video embeddings (Fig. 7). The decoder predicts the next word based on word vector w_t at each time step t as:

$$[a_t \ i_t \ f_t \ o_t]^T = W_u w_t + b_u + W_1 h_{t-1}, \quad (11)$$

$$c_t = \tanh(a_t)\sigma(i_t) + c_{t-1}\sigma(f_t), \quad (12)$$

$$h_t = \tanh(c_t)\sigma(o_t), \quad (13)$$

$$p_t = \text{softmax}(W_p h_t + b_p) \quad (14)$$

where $W_u, W_1 \in \mathbb{R}^{4d_w \times d_w}$ and $b_u \in \mathbb{R}^{4d_w}$ are parameters of the LSTM, and $[a_t \ i_t \ f_t \ o_t]^T$ is a column vector that is a concatenation of $a_t, i_t, f_t, o_t \in \mathbb{R}^{d_w}$. The matrix W_p and the vector b_p encode the hidden state into a vector with the vocabulary size. The output p_t is the probabilities over the vocabulary. We built a vocabulary consisting of all words in the YouTube dataset and special tags, *i.e.*, begin-of-sentence (“<bos >”) and end-of-sentence (“<eos >”). The generative process is terminated when “<eos >” is produced. We trained the decoder using the YouTube dataset. We computed the video embedding $\phi_v(X)$ using GoogLeNet+ALL₂ as an input to the LSTM at $t = 0$. We trained the decoder by minimizing the cross entropy loss. During training, we fixed the parameters of our embedding models.

Figure 8 shows generated sentences. Although video embeddings were trained for retrieval tasks and not finetuned for the decoder, we observed that most generated sentences were semantically relevant to their original videos.

We evaluated generated sentences with the COCO description evaluation. We found that the scores were comparable to prior work (Table 3). This indicates that our model efficiently encoded videos, maintaining their semantics. Moreover, this result suggests that our embeddings can be applied to other tasks that require joint representations of videos and sentences.

6 Conclusion

We presented a video and sentence retrieval framework that incorporates web images to bridge between sentences and videos. Specifically, we collected web image search results in order to disambiguate semantics of a sentence. We developed neural network-based embedding models for video, sentence, and image inputs which fuses sentence and image representations. We jointly trained video and sentence embeddings using the YouTube dataset. Our experiments demonstrated the advantage of incorporating additional web images, and our approach clearly outperformed prior work in the both video and sentence retrieval tasks. Furthermore, by decoding descriptions from video embeddings, we demonstrated that rich semantics of videos were efficiently encoded in our video embeddings. Our future work includes developing a video embedding that considers temporal structures of videos. It would be also interesting to investigate what kind of sentences benefit from image search results, and how to collect efficient images.

References

1. Chen, D.L., Dolan, W.B.: Collecting highly parallel data for paraphrase evaluation. In: ACL. pp. 190–200 (2011)
2. Chen, X., Fang, H., Lin, T., Vedantam, R., Gupta, S., Dollr, P., Zitnick, C.L.: Microsoft COCO Captions: Data collection and evaluation server. arXiv preprint arXiv:1504.00325 p. 7 pages (2015)
3. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: CVPR. pp. 539–546 (2005)
4. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys* 40(2), 5:1–5:60 (2008)
5. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: DeCAF: A deep convolutional activation feature for generic visual recognition. In: ICML. pp. 647–655 (2014)
6. Fang, H., Gupta, S., Iandola, F., Srivastava, R.K., Deng, L., Dollar, P., Gao, J., He, X., Mitchell, M., Platt, J.C., Zitnick, C.L., Zweig, G.: From captions to visual concepts and back. In: CVPR. pp. 1473–1482 (2015)
7. Farhadi, A., Hejrati, M., Sadeghi, M.A., Young, P., Rashtchian, C., Hockenmaier, J., Forsyth, D.: Every picture tells a story: Generating sentences from images. In: ECCV. pp. 15–29 (2010)
8. Fergus, R., Fei-Fei, L., Perona, P., Zisserman, A.: Learning object categories from Google’s image search. In: ICCV. pp. 1816–1823 (2005)
9. Frome, A., Corrado, G.S., Shlens, J., Bengio, S., Dean, J., Ranzato, M.A., Mikolov, T.: DeViSE: A deep visual-semantic embedding model. In: NIPS. pp. 2121–2129 (2013)
10. Girshick, R., Donahue, J., Darrell, T., Berkeley, U.C., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR. pp. 580–587 (2014)
11. Guadarrama, S., Venugopalan, S., Austin, U.T., Krishnamoorthy, N., Mooney, R., Malkarnenkar, G., Darrell, T., Berkeley, U.C.: YouTube2Text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In: ICCV. pp. 2712–2719 (2013)
12. Gygli, M., Grabner, H., Van Gool, L.: Video summarization by learning submodular mixtures of objectives. In: CVPR. pp. 3090–3098 (2015)
13. Johnson, J., Ballan, L., Fei-Fei, L.: Love thy neighbors: Image annotation by exploiting image metadata. In: ICCV. pp. 4624 – 4632 (2015)
14. Karpathy, A., Joulin, A., Fei-Fei, L.: Deep fragment embeddings for bidirectional image sentence mapping. In: NIPS, pp. 1889–1897 (2014)
15. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: ICLR. p. 11 pages (2015)
16. Kiros, R., Zhu, Y., Salakhutdinov, R.R., Zemel, R., Urtasun, R., Torralba, A., Fidler, S.: Skip-thought vectors. In: NIPS. pp. 3276–3284 (2015)
17. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning (ICML). pp. 1188–1196 (2014)
18. Lin, D., Fidler, S., Kong, C., Urtasun, R.: Visual semantic search: Retrieving videos via complex textual queries. In: CVPR. pp. 2657–2664 (2014)
19. Lin, T.Y., Belongie, S., Hays, J.: Learning deep representations for ground-to-aerial geolocalization. In: CVPR. pp. 5007–5015 (2015)
20. Maybank, S.: A survey on visual content-based video indexing and retrieval. *IEEE Trans. Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 41(6), 797–819 (2011)

21. Ordonez, V., Kulkarni, G., Berg, T.: Im2Text: Describing images using 1 million captioned photographs. In: NIPS. pp. 1143–1151 (2011)
22. Rashtchian, C., Young, P., Hodosh, M., Hockenmaier, J.: Collecting image annotations using amazon’s mechanical turk. In: NAACL-HLT. pp. 139–147 (2010)
23. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS, pp. 91–99 (2015)
24. Rohrbach, M., Qiu, W., Titov, I., Thater, S., Pinkal, M., Schiele, B.: Translating video content to natural language descriptions. In: ICCV. pp. 433–440 (2013)
25. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* 115(3), 211–252 (2015)
26. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR. p. 14 pages (2015)
27. Socher, R., Ganjoo, M., Manning, C.D., Ng, A.Y.: Zero-shot learning through cross-modal transfer. In: NIPS. pp. 935–943 (2013)
28. Song, Y., Vallmitjana, J., Stent, A., Jaimes, A.: TVSum : Summarizing web videos using titles. In: CVPR. pp. 5179–5187 (2015)
29. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR. pp. 1–9 (2015)
30. Tokui, S., Oono, K., Hido, S., Clayton, J.: Chainer: A next-generation open source framework for deep learning. In: NIPS. p. 6 pages (2015)
31. Venugopalan, S., Xu, H., Donahue, J., Rohrbach, M., Mooney, R., Saenko, K.: Translating videos to natural language using deep recurrent neural networks. In: NAACL-HLT. pp. 1494–1504 (2014)
32. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. In: CVPR. pp. 3156–3164 (2015)
33. Wang, X., Gupta, A.: Unsupervised learning of visual representations using videos. In: ICCV. pp. 2794–2802 (2015)
34. Xu, R., Xiong, C., Chen, W., Corso, J.: Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In: AAAI. pp. 2346–2352 (2015)
35. Yao, L., Ballas, N., Larochelle, H., Courville, A.: Describing videos by exploiting temporal structure. In: ICCV. pp. 4507 – 4515 (2015)
36. Young, P., Lai, A., Hodosh, M., Hockenmaier, J.: From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics* 2, 67–78 (2014), <https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/229>
37. Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., Fidler, S.: Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In: IEEE International Conference on Computer Vision (ICCV). pp. 19–27 (2015)