Dual Encoding for Zero-Example Video Retrieval

Jianfeng Dong[1], Xirong Li*[2,3], Chaoxi Xu[3], Shouling Ji[4,5], Yuan He[6], Gang Yang[3], and Xun Wang[1]

[1]College of Computer and Information Engineering, Zhejiang Gongshang University
[2]Key Lab of Data Engineering and Knowledge Engineering, Renmin University of China
[3]AI & Media Computing Lab, School of Information, Renmin University of China
[4]College of Computer Science, Zhejiang University
[5]Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies
[6]Alibaba Group

Abstract

This paper attacks the challenging problem of zero-example video retrieval. In such a retrieval paradigm, an end user searches for unlabeled videos by ad-hoc queries described in natural language text with no visual example provided. Given videos as sequences of frames and queries as sequences of words, an effective sequence-to-sequence cross-modal matching is required. The majority of existing methods are concept based, extracting relevant concepts from queries and videos and accordingly establishing associations between the two modalities. In contrast, this paper takes a concept-free approach, proposing a dual deep encoding network that encodes videos and queries into powerful dense representations of their own. Dual encoding is conceptually simple, practically effective and end-to-end. As experiments on three benchmarks, i.e. *MSR-VTT, TRECVID 2016 and 2017 Ad-hoc Video Search show, the proposed solution establishes a new state-of-the-art for zero-example video retrieval.*

1. Introduction

This paper targets at *zero-example* video retrieval, where a query is described in natural language text and no visual example is given. The topic is fundamentally interesting as it requires establishing proper associations between visual and linguistic information presented in temporal order.

Zero-example video retrieval attracts initial attention in the form of zero-example multimedia event detection, where the goal is to retrieve video shots showing specific events such as *parking a vehicle*, *dog show* and *birthday party*, but with no training videos provided [3, 6, 9, 12, 20, 31]. All these methods are concept based, *i.e*. describing the video content by automatically detected concepts, which are used to match with a target event. Such a concept-based tradition continues. For the NIST TRECVID challenge of zero-example video retrieval [2], we observe that the top performers are mostly concept based [15, 22, 25, 30]. However, the concept-based paradigm faces a number of difficulties including how to specify a set of concepts, how to train good classifiers for these concepts, and more cru-

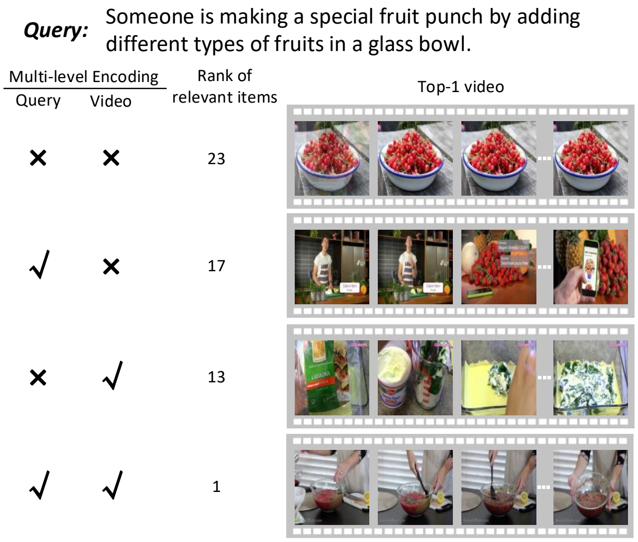
Query: Someone is making a special fruit punch by adding different types of fruits in a glass bowl.
Multi-level Encoding
Query
Video
Rank of relevant items
Top-1 video
× × 23
√ × 17
× √ 13
√ √ 1

Figure 1. **Showcase of zero-example video retrieval with and without the proposed encoding**. The ✖ symbol indicates encoding by mean pooling. Numbers in the third column are the rank of the relevant video returned by retrieval models subject to specific query / video encoding strategies. The retrieval model with dual encoding successfully answers this complex query.

*Xirong Li is the corresponding author (xirong@ruc.edu.cn).

cially how to select relevant and detectable concepts for both video and query representation [20]. This paper differs from these works as we aim for a concept-free method that learns directly common semantic embeddings for both videos and queries.

Good efforts have been made for learning joint embeddings of the two modalities for zero-example video retrieval [24, 34]. In [34], a query sentence is vectorized by a recursive neural network, while [24] vectorizes a given sentence by a recurrent neural network. In both works a specific video is vectorized by mean pooling of visual features of its frames. Different from [24,34], we propose *dual multi-level encoding* for both videos and queries in advance to common space learning. As exemplified in Figure 1, the new encoding strategy is crucial for describing complex queries and video content.

Our hypothesis is that a given video / query has to be first encoded into a powerful representation of its own. We consider such a decomposition crucial as it allows us to design an encoding network that jointly exploits multiple encoding strategies including mean pooling, recurrent neural networks and convolutional networks. In our design, the output of a specific encoding block is not only used as input of a follow-up encoding block, but also re-used via skip connections to contribute to the final output. It generates new, higher-level features progressively. These features, generated at distinct levels, are powerful and complementary to each other, allowing us to obtain effective video (and sentence) representations by very simple concatenation. Moreover, dual encoding is orthogonal to common space learning, allowing us to flexibly embrace state-of-the-art common space learning algorithms.

In sum, this paper makes the following contributions.

- We propose multi-level encodings of video and text in advance to learning shared representations for the two modalities. As such, the encodings are not meant for direct video-text matching. This is conceptually different from existing works that tackle cross-modality matching as a whole.
- We design a novel *dual* network that encodes an input, let it be a query sentence or a video, in a similar manner. By jointly exploiting multi-level encodings, the network explicitly and progressively learns to represent global, local and temporal patterns in videos and sentences. While being orthogonal to common space learning, the entire model is trained in an end-to-end manner.
- Dual encoding, combined with state-of-the-art common space learning [8], establishes a new state-of-the-art for zero-example video retrieval, as supported by its superior performance on three challenging benchmarks. Code and data are available at https://github.com/danieljf24/dual_encoding.

2. Related Work

Since 2016 the TRECVID starts a new challenge for zero-example video retrieval, known as Ad-hoc Video Search (AVS) [2]. The majority of the top ranked solutions for this challenge depend on visual concept classifiers to describe video content and linguistic rules to detect concept in textual queries [15, 21, 22, 25, 30]. For instance, [21, 22] utilize multiple pre-trained Convolutional Neural Network (CNN) models to detect main objects and scenes in video frames. As for query representation, the authors design relatively complex linguistic rules to extract relevant concepts from a given query. Ueki *et al*. [30] come with a much larger concept bank consisting of more than 50k concepts. In addition to pre-trained CNN models, they train SVM classifiers to automatically annotate the video content. We argue that such a concept-based paradigm has a fundamental disadvantage. That is, it is very difficult, if not impossible, to describe the rich sequential information within both video and query using a few selected concepts.

With big advances of deep learning in natural language processing and computer vision research, we observe an increased use of such techniques for video retrieval [7, 24, 34, 36, 37]. By directly encoding videos and text into a common space, these methods are concept free. For video encoding, a typical approach is to first extract visual features from video frames by pre-trained CNN models, and subsequently aggregate the frame-level features into a video-level feature. To that end, mean pooling is the *de facto* choice [7,24,34]. To explicitly model the temporal information, Yu *et al*. [37] use Long Short-Term Memory (LSTM), where frame-level features are sequentially fed into LSTM, and the hidden vector at the last step is used as the video feature. CNN is exploited in [36]. None of the above works considers more than one video encoding strategy.

For query sentence encoding, while bag-of-words remains popular [10], deep networks are in increasing use. Recursive neural networks are investigated in [34] for vectorizing subject-verb-object triplets extracted from a given sentence. Variants of recurrent neural networks are being exploited, see the usage of LSTM, bidirectional LSTM, and Gated Recurrent Unit (GRU) in [37], [36], and [24], respectively. To the best of our knowledge, [7] is the only work looking to a joint use of multiple sentence encoding strategies including bag-of-words, word2vec and GRU. However, as aforementioned, that work simply uses mean pooling for video encoding.

To the best of our knowledge, this paper is the first work for explicitly and progressively exploiting global, local and temporal patterns in both videos and sentences.

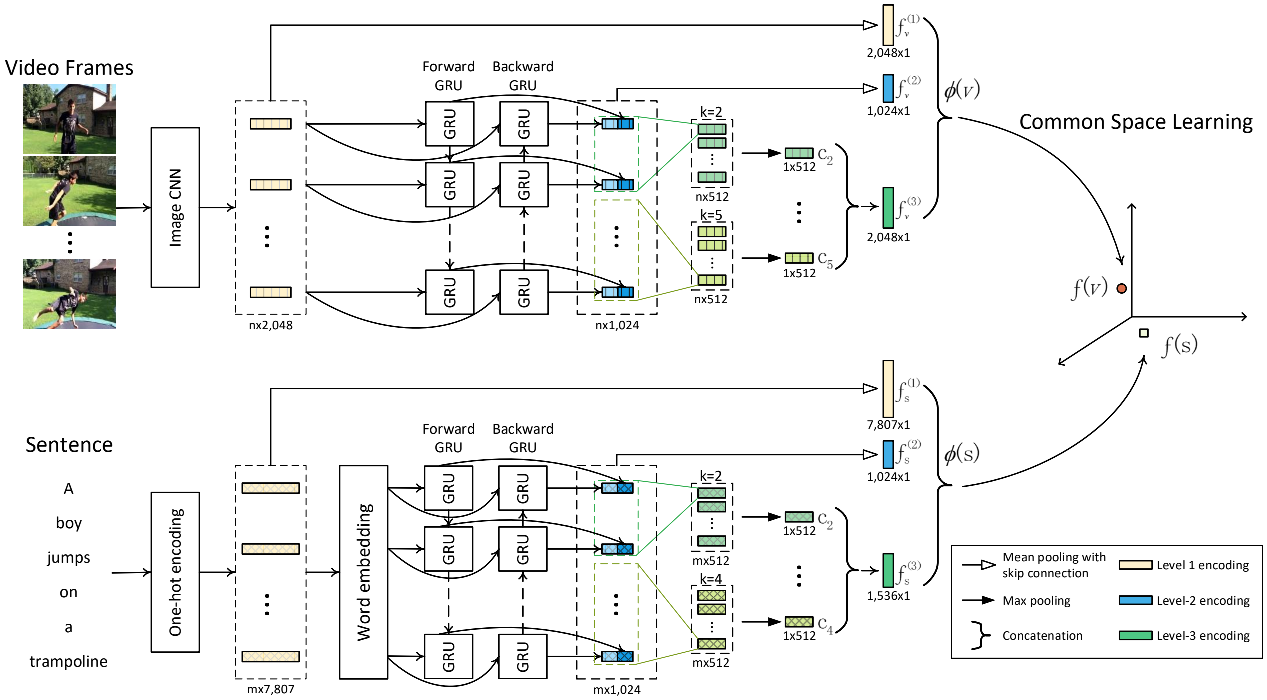

Figure 2. **A conceptual diagram of the proposed dual encoding network for zero-example video retrieval**. Given a video v and a sentence s, the network performs in parallel multi-level encodings, *i.e.* mean pooling, biGRU and biGRU-CNN, eventually representing the two input by two combined vectors $\phi(v)$ and $\phi(s)$, respectively. The vectors are later projected into a common space, which we learn using VSE++ [8], for video-text similarity computation. Note that the length of the bag-of-words vector $f_s^{(1)}$ is equal to the size of the training vocabulary, which is 7,807 when we train on MSR-VTT. Once the network is trained, encoding at each side is performed independently, meaning we can process large-scale videos offline and answer ad-hoc queries on the fly.

3. The *Dual Encoding* Network

Given a video v and a sentence s, we propose to encode them in a dual manner, denoted as $\phi(v)$ and $\phi(s)$, in advance to common space learning. As illustrated in Figure 2, multi-level encodings are performed for each modality. The encoding results are combined to describe the two modalities in a coarse-to-fine fashion. Both video and sentence are essentially a sequence of items, let it be frames or words. Such a property allows us to design a dual encoding network to handle the two distinct modalities. In what follows we first depict the network at the video side. We then specify choices that are unique to text.

3.1. Video-side Multi-level Encoding

For a given video, we extract uniformly a sequence of n frames with a pre-specified interval of 0.5 second. Per frame we extract deep features using a pretrained ImageNet CNN, as commonly used for video content analysis [10, 21, 34]. Consequently, the video is described by a sequence of feature vectors $\{v_1, v_2, \dots, v_n\}$, where v_t indicates the deep feature vector of the t-th frame. Notice that 3D CNNs [29] can also be used for feature extraction when treating segments of frames as individual items.

3.1.1 Level 1. Global Encoding by Mean Pooling

According to our literature review, mean pooling, which represents a video by simply averaging the features of its frames, is arguably the most popular choice for zero-example video retrieval. By definition, mean pooling captures visual patterns that repeatedly present in the video content. These patterns tend to be global. We use $\bar{v}_v$ to indicate the encoding result at this level,

$$f_v^{(1)} = \frac{1}{n}\sum_{t=1}^{n} v_t. \tag{1}$$

3.1.2 Level 2. Temporal-Aware Encoding by biGRU

Bi-directional recurrent neural network [27] is known to be effective for making use of both past and future contextual information of a given sequence. We hypothesize that such a network is also effective for modeling the video temporal information. We adopt a bidirectional GRU (biGRU) [5], which has less parameters than the bidirectional LSTM and thus requires less amounts of training data. A biGRU consists of two separated GRU layers, *i.e.* a forward GRU and a backward GRU. The forward GRU is used to encode frame features in normal order, while the backward

GRU encodes frame features in reverse order. Let $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ be their corresponding hidden states at a specific time step $t = 1, \ldots, n$. The hidden states are generated as

$$\begin{aligned}\overrightarrow{h}_t &= \overrightarrow{GRU}(v_t, \overrightarrow{h}_{t-1}), \\ \overleftarrow{h}_t &= \overleftarrow{GRU}(v_{n+1-t}, \overleftarrow{h}_{t-1}),\end{aligned} \tag{2}$$

where $\overrightarrow{GRU}$ and $\overleftarrow{GRU}$ indicate the forward and backward GRUs, with past information carried by $\overrightarrow{h}_{t-1}$ and $\overrightarrow{h}_{t-1}$, respectively. Concatenating $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$, we obtain the biGRU output $h_t = [\overrightarrow{h}_t, \overleftarrow{h}_t]$. The size of the hidden vectors in the forward and backward GRUs is empirically set to 512. Accordingly, the size of h_t is 1,024. Putting all the output together, we obtain a feature map $H = \{h_1, h_2, ..., h_n\}$, with a size of $1,024 \times n$. The biGRU based encoding, denoted $\bar{h}_v$, is obtained by applying mean pooling on H along the row dimension, that is

$$f_v^{(2)} = \frac{1}{n} \sum_{t=1}^{n} h_t. \tag{3}$$

3.1.3 Level 3. Local-Enhanced Encoding by biGRU-CNN

The previous layer treats the output of biGRU at each step equally. To enhance local patterns that help discriminate between videos of subtle difference, we build convolutional networks on top of biGRU. In particular, we adapt 1-d CNN originally developed for sentence classification [13].

The input of our CNN is the feature map H generated by the previous biGRU module. Let $Conv1d_{k,r}$ be a 1-d convolutional block that contains $r = 512$ filters of size k, with $k \geq 2$. Feeding H, after zero padding, into $Conv1d_{k,r}$ produces a $n \times r$ feature map. Non-linearity is introduced by applying the ReLU activation function on the feature map. As n varies for videos, we further apply max pooling to compress the feature map to a vector c_k of fixed length r. More formally we express the above process as

$$c_k = \text{max-pooling}(ReLU(Conv1d_{k,r}(H))). \tag{4}$$

A filter with $k = 2$ allows two adjacent rows in H to interact with each other, while a filter of larger k means more adjacent rows are exploited simultaneously. In order to generate a multi-scale representation, we deploy multiple 1-d convolutional blocks with $k = 2, 3, 4, 5$. Their output is concatenated to form the biGRU-CNN based encoding, *i.e.*

$$f_v^{(3)} = [c_2, c_3, c_4, c_5]. \tag{5}$$

As $f_v^{(1)}$, $f_v^{(2)}$, $f_v^{(3)}$ are obtained sequentially at different levels by specific encoding strategies, we consider it reasonable to presume that the three encoding results are complementary to each other, with some redundancy. Hence, we obtain multi-level encoding of the input video by concatenating the output from all the three levels, namely

$$\phi(v) = [f_v^{(1)}, f_v^{(2)}, f_v^{(3)}]. \tag{6}$$

In fact, this concatenation operation, while being simple, is a common practice for feature combination [11, 38].

3.2. Text-side Multi-level Encoding

The above encoding network, after minor modification, is also applicable for text.

Given a sentence s of length m, we represent each of its words by a one-hot vector. Accordingly, a sequence of one-hot vectors $\{w_1, w_2, \ldots, w_m\}$ is generated, where w_t indicates the vector of the t-th word. Global encoding $f_s^{(1)}$ is obtained by averaging all the individual vectors in the sequence. This amounts to the classical bag-of-words representation.

For biGRU based encoding, each word is first converted to a dense vector by multiplying its one-hot vector with a word embedding matrix. We initialize the matrix using a word2vec [23] model provided by [7], which trained word2vec on English tags of 30 million Flickr images. The rest is mostly identical to the video counterpart. We denote the biGRU based encoding of the sentence as $f_s^{(2)}$. Similarly, we have the biGRU-CNN based encoding of the sentence as $f_s^{(3)}$. Here, we utilize three 1-d convolutional blocks with $k = 2, 3, 4$. Multi-level encoding of the sentence is obtained by concatenating the encoding results from all the three levels in the dual network, *i.e.*

$$\phi(s) = [f_s^{(1)}, f_s^{(2)}, f_s^{(3)}]. \tag{7}$$

As $\phi(v)$ and $\phi(s)$ have not been correlated, they are not directly comparable. For video-text similarity computation, the vectors need to be projected into a common space, the learning algorithm for which will be presented next.

4. Common Space Learning

Among many choices of common space learning algorithms we choose VSE++ [8] for two reasons. First, it is the state-of-the-art in its original context of image-text retrieval, and more recently found to be effective also in the video domain [24]. Second, its source code is publicly available[1], which greatly facilitates our exploitation of the algorithm.

Given the encoded video vector $\phi(v)$ and sentence vector $\phi(s)$, we project them into a common space by affine transformation. From the neural network viewpoint, affine transformation is essentially a Fully Connected (FC) layer. On the basis of [8], we additionally use a Batch Normalization (BN) layer after the FC layer, as we find this trick beneficial. Putting everything together, we obtain the video

[1] https://github.com/fartashf/vsepp

feature vector $f(v)$ and sentence feature vector $f(s)$ in the common space as

$$\begin{aligned} f(v) &= \text{BN}(W_v \phi(v) + b_v), \\ f(s) &= \text{BN}(W_s \phi(s) + b_s), \end{aligned} \tag{8}$$

where W_v and W_s parameterize the FC layers on each side, with b_v and b_s as bias terms.

The dual encoding network and the common space learning network are trained together in an end-to-end manner except that image convnets used for extracting video features are pre-trained and fixed. Let θ be all the trainable parameters. The video-text similarity subject to θ, denoted by $S_\theta(v, s)$, is computed using cosine similarity[2] between $f(v)$ and $f(s)$.

We use the improved marginal ranking loss [8], which penalizes the model according to the hardest negative examples. Concretely, the loss $\mathcal{L}(v, s; \theta)$ for a relevant video-sentence pair is defined as

$$\begin{aligned} \mathcal{L}(v, s; \theta) = max(0, \alpha + S_\theta(v, s^-) - S_\theta(v, s)) \\ + max(0, \alpha + S_\theta(v^-, s) - S_\theta(v, s)), \end{aligned} \tag{9}$$

where α is the margin constant, while s^- and v^- respectively indicate a negative sentence sample for v and a negative video sample for s. The two negatives are not randomly sampled. Instead, the most similar yet negative sentence and video in the current mini-batch are chosen. The entire network is trained towards minimizing this loss.

5. Evaluation

We conduct five experiments. First, following [24], we perform text-to-video and video-to-text retrieval on the MSR-VTT dataset [33]. We then evaluate the proposed method in the context of the TRECVID Ad-hoc Video Search task of the last two years [1, 2]. Further, we evaluate on MSVD [4] for cross-dataset generalization and MPII-MD [26] for cross-domain generalization. While focusing on video retrieval, we provide an additional experiment on Flickr30k [35] and MS-COCO [19] to investigate if the VSE++ model [8], the state-of-the-art for image-text retrieval, can be improved by replacing its GRU based encoding by the proposed encoding at the text side. Lastly, for ad-hoc video retrieval where a user submits queries on the fly, retrieval speed matters. So an efficiency test is provided.

Before proceeding to the experiments, we detail our implementations. We use PyTorch (http://pytorch.org) as our deep learning environment. For sentence preprocessing, we first convert all words to the lowercase and then replace words that occurring less than five times in the training set with a special token. We empirically set the size of the learned common space to 2,048, and the margin parameter α to 0.2. We use SGD with Adam [14]. The mini-batch size is 128. With an initial learning rate of 0.0001, we take an adjustment schedule similar to [7]. That is, 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 50.

5.1. Experiments on MSR-VTT

Setup. The MSR-VTT dataset [33], originally developed for video captioning, consists of 10k web video clips and 200k natural sentences describing the visual content of the clips. The average number of sentences per clip is 20. We use the official data partition, *i.e.* 6,513 clips for training, 497 clips for validation, and the remaining 2,990 clips for testing.

For method comparison, we consider [24], the first work reporting video retrieval performance on MSR-VTT. A more recent work [36] also experiments with MSR-VTT, but uses a non-public subset, making its results not comparable. We include W2VV [7], another state-of-the-art model with code available[3]. W2VV uses the Mean Square Error (MSE) loss. So for a fair comparison, we adapt the model by substituting the improved marginal ranking loss for MSE and train it from scratch. We term this variant as W2VV$_{\text{imrl}}$. The same 2,048-dim ResNet-152 feature as [24] is used.

We report rank-based performance metrics, namely $R@K$ ($K = 1, 5, 10$), Median rank (Med r) and mean Average Precision (mAP). $R@K$ is the percentage of test queries for which at least one relevant item is found among the top-K retrieved results. Med r is the median rank of the first relevant item in the search results. Higher $R@K$, mAP and lower Med r mean better performance. For overall comparison, we report the sum of all recalls. Note that for text-to-video retrieval, each test sentence is associated with one relevant video, while for video-to-text retrieval, each test video is associated with 20 relevant sentences. So the latter will have better performance scores.

Comparison with the State-of-the-art. Table 1 shows the performance on MSR-VTT. Though our goal is zero-example video retrieval, which corresponds to text-to-video retrieval in the table, video-to-text retrieval is also included for completeness. While [7] is less effective than [24], letting the former use a similar loss function as the latter brings in a considerable performance gain, with the sum of recalls increased from 90.3 to 132.1. The result suggests the importance of assessing different video / text encoding strategies within the same common space learning framework. The proposed method performs the best.

Ablation Study. To exam the usefulness of each component in the dual encoding network, we conduct an ab-

[2] In our preliminary experiment, we also tried the Euclidean distance, but found it less effective.

[3] https://github.com/danieljf24/w2vv

Table 1. **State-of-the-art on MSR-VTT**. Larger R@{1,5,10}, mAP and smaller Med r indicate better performance. Methods sorted in ascending order in terms of their overall performance. The proposed method performs the best.

Method	Text-to-Video Retrieval					Video-to-Text Retrieval					Sum of Recalls
	R@1	R@5	R@10	Med r	mAP	R@1	R@5	R@10	Med r	mAP	
W2VV [7]	1.8	7.0	10.9	193	0.052	9.2	25.4	36.0	24	0.050	90.3
Mithun *et al.* [24]	5.8	17.6	25.2	61	-	10.5	26.7	35.9	25	-	121.7
W2VV$_{\text{imrl}}$	6.1	18.7	27.5	45	0.131	11.8	28.9	39.1	21	0.058	132.1
Dual encoding	**7.7**	**22.0**	**31.8**	**32**	**0.155**	**13.0**	**30.8**	**43.3**	**15**	**0.065**	**148.6**

Table 2. **Ablation study on MSR-VTT**. The overall performance, as indicated by **Sum of Recalls**, goes up as more encoding layers are added. Dual encoding exploiting all the three levels is the best.

Encoding strategy	Text-to-Video Retrieval					Video-to-Text Retrieval					Sum of Recalls
	R@1	R@5	R@10	Med r	mAP	R@1	R@5	R@10	Med r	mAP	
Level 1 (Mean pooling)	6.4	18.8	27.3	47	0.132	11.5	27.7	38.2	22	0.054	129.9
Level 2 (biGRU)	6.3	19.4	28.5	38	0.136	10.1	26.8	37.7	20	0.057	128.8
Level 3 (biGRU-CNN)	7.3	21.5	31.2	32	0.150	10.6	27.3	38.5	20	0.061	136.4
Level 1 + 2	6.9	20.4	29.1	41	0.142	11.6	29.6	40.7	18	0.058	138.3
Level 1 + 3	7.5	21.6	31.2	33	0.151	11.9	30.5	41.7	16	0.062	144.4
Level 2 + 3	7.6	**22.4**	**32.2**	**31**	**0.155**	11.9	**30.9**	42.7	16	**0.066**	147.7
Level 1 + 2 + 3	**7.7**	22.0	31.8	32	**0.155**	**13.0**	30.8	**43.3**	**15**	0.065	**148.6**

lation study as follows. Given varied combinations of the components, seven models are trained. Table 2 summarizes the choices of video and text encodings and the corresponding performance. Among the individual encoding levels, biGRU-CNN, which builds CNN on top of the output of biGRU turns out to be the most effective. As more encoding layers are included, the overall performance goes up. For the last four models which combines output from previous layers, they all outperform the first three models. This suggests that different layers are complementary to each other. The full multi-level encoding setup, *i.e.* Level 1 +2 + 3 in Table 2, is the best.

We also investigate single-side encoding, that is, video-side multi-level encoding with mean pooling on the text side and text-side multi-level encoding with mean pooling on the video side. These two strategies obtain Sum of Recalls of 143.6 and 137.1, respectively. The lower scores justify the necessity of dual encoding. The result also suggests that video-side encoding is more beneficial.

Table 3. **State-of-the-art on TRECVID 2016**.

Method	infAP
Top-3 TRECVID finalists:	
Le *et al.* [15]	0.054
Markatopoulou *et al.* [22]	0.051
Liang *et al.* [18]	0.040
Literature methods:	
Habibian *et al.* [10]	0.087
Markatopoulou *et al.* [21]	0.064
W2VV$_{\text{imrl}}$	0.132
Dual encoding	**0.159**

Table 4. **State-of-the-art on TRECVID 2017**.

Method	infAP
Top-3 TRECVID finalists:	
Snoek *et al.* [28]	0.206
Ueki *et al.* [30]	0.159
Nguyen *et al.* [25]	0.120
Literature methods:	
Habibian *et al.* [10]	0.150
W2VV$_{\text{imrl}}$	0.165
Dual encoding	**0.208**

5.2. Experiments on TRECVID

Setup. We evaluate dual encoding in the TRECVID AVS task [1, 2], which provides the largest test bed for zero-example video retrieval to this date. The test collection, called IACC.3, contains 4,593 Internet Archive videos with duration ranging from 6.5 min to 9.5 min and a mean duration of almost 7.8 min. Shot boundary detection results in 335,944 shots in total. Given an ad-hoc query, *e.g. Find shots of military personnel interacting with protesters*, the task is to return for the query a list of 1,000 shots from the test collection ranked according to their likelihood of containing the given query. Per year TRECVID specifies 30 distinct queries of varied complexity.

As TRECVID does not specify training data for the AVS task, we train the dual encoding network using the joint collection of MSR-VTT and the TGIF [17], which contains 100K animated GIFs and 120K sentences describing visual content of the GIFs. Although animated GIFs are a very different domain, TGIF was constructed in a way to resemble user-generated video clips, *e.g.* with cartoon, static, and textual content removed. For IACC.3, MSR-VTT and TGIF, we use frame-level CNN features provided by [16], where the authors use ResNeXt-101 [32] trained on the full ImageNet collection for feature extraction.

For method comparison, we include the top 3 entries of each year, *i.e.* [15,18,22] for 2016 and [25,28,30] for 2017. Besides we include publications that report performance on the tasks, *i.e.* [10, 21], to the best of our knowledge. As noted in our literature review, most of the methods are concept based. Notice that visual features and training data used by these methods vary, meaning the comparison and consequently conclusions drawn from this comparison is at a system level. So for a more conclusive comparison, we re-train W2VV$_{\text{imrl}}$ using the same joint dataset and the same ResNeXt-101 feature.

We report inferred Average Precision (infAP), the official performance metric used by the TRECVID AVS task. The overall performance is measured by averaging infAP scores over the queries.

Comparison with the State-of-the-art Table 3 and 4 show the performance of different methods on the TRECVID 2016 and 2017 AVS tasks, respectively. The proposed method again performs the best, with infAP of 0.159 and 0.208. While [28] has a close infAP of 0.206 on the TRECVID 2017 task, their solution ensembles ten models. Their best single model, *i.e.* [10] which uses the same ResNeXt-101 feature, has a lower infAP of 0.150. Given the same training data and feature, the proposed method outperforms W2VV$_{\text{imrl}}$ as well. Table 5 shows the influence of distinct training data. The above results again justify the effectiveness of dual encoding.

Note that the TRECVID ground truth is partially available at the shot-level. The task organizers employ a pooling strategy to collect the ground truth, *i.e.* a pool of candidate shots are formed by collecting the top-1000 shots from each submission and a random subset is selected for manual verification. The ground truth thus favors official participants. As the top ranked items found by our method can be outside of the subset, infAP scores of our method is likely to be underestimated. Some qualitative results are show in Fig. 3.

5.3. Experiments on MSVD and MPII-MD

Setup. We evaluate on MSVD [4] and MPII-MP [26], two other popular video sets. Note that MSR-VTT is built in a similar vein to MSVD , but in a larger scale. So we assess the models previously trained on MSR-VTT using the MSVD test set. MPII-MD, as a movie description dataset, is unique. So we re-train and evaluate all the models on this dataset with its official data split. The ResNeXt-101 feature is used.

Table 5. **Performance of our model trained on distinct data for the TRECVID AVS benchmarks**. Performance metric: infAP.

Training data	TRECVID 2016	TRECVID 2017
MSR-VTT	0.140	0.179
TGIF	0.092	0.199
MSR-VTT + TGIF	**0.159**	**0.208**

Table 6. **Performance of zero-example video retrieval, measured by mAP**. Our proposed method is the best.

Model	MSVD	MPII-MD
W2VV	0.100	0.008
W2VV$_{\text{imrl}}$	0.230	0.030
VSE++	0.218	0.022
Dual Encoding	**0.232**	**0.037**

Table 7. **Performance of image-text retrieval on Flickr30k and MSCOCO**. The proposed multi-level encoding is beneficial for the VSE++ model [8].

Method	Text-to-Image			Image-to-Text		
	R@1	R@5	R@10	R@1	R@5	R@10
On Flickr30k						
VSE++	23.1	49.2	60.7	31.9	58.4	68.0
VSE++, multi-level encoding	**24.7**	**52.3**	**65.1**	**35.1**	**62.2**	**71.3**
On MSCOCO						
VSE++	33.7	68.8	81.0	43.6	74.8	84.6
VSE++, multi-level encoding	**34.8**	**69.6**	**82.6**	**46.7**	**76.2**	**85.8**

Results. As Table 6 shows, our model again performs the best in the cross-dataset scenario. Our model is also the most effective on MPII-MP. See Fig. 4 for qualitative results of zero-example movie retrieval.

5.4. Experiments on Flickr30K and MSCOCO

Setup. We investigate if the VSE++ model [8] can be improved in its original context of image-text retrieval, when replacing its textual encoding module, which is a GRU, by the proposed multi-level encoding module. To that end, we fix all other choices, adopting the exact evaluation protocol of [8]. That is, we use the same data split, where the training / validation / test test has 30,000 / 1,000 / 1,000 images for Flickr30K, and 82,783 / 5,000 / 5,000 images for MSCOCO. We also use the same VGGNet feature provided by [8]. Performance of $R@1$, $R@5$ and $R@10$ are reported. On MSCOCO, the results are reported by averaging over 5 folds of 1,000 test images.

Results. Table 7 shows the performance of image-text retrieval on Flickr30k and MSCOCO. Integrating text-side multi-level encoding into VSE++ brings improvements on both datasets. The results suggest that the proposed text-

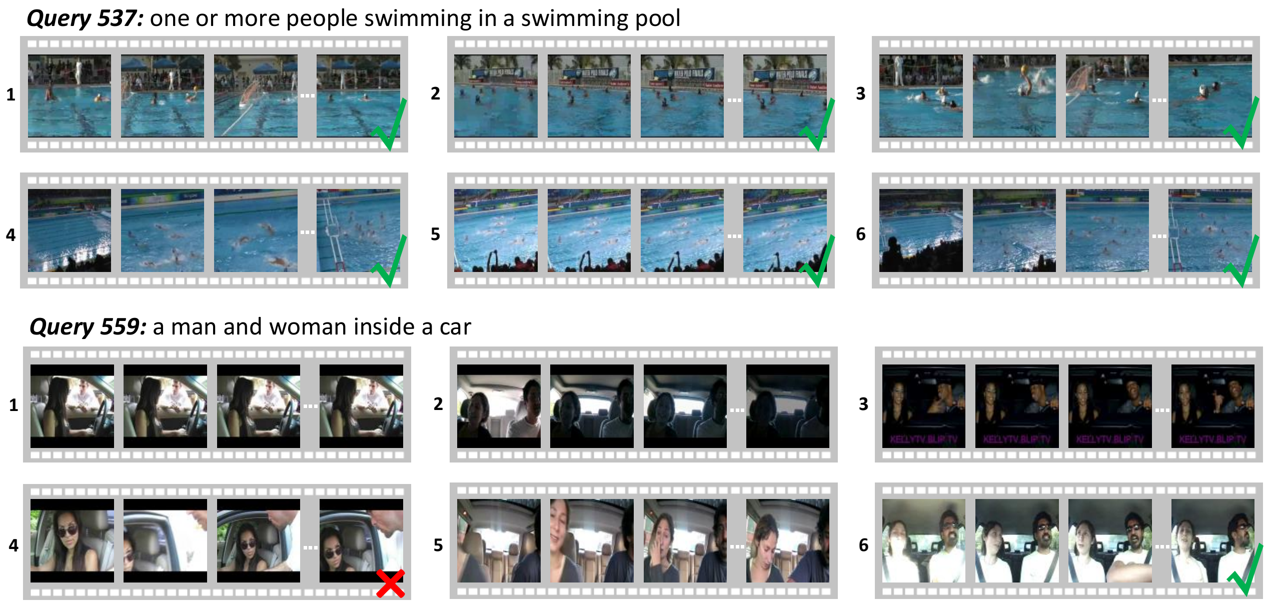
Query 537: one or more people swimming in a swimming pool
1
2
3
4
5
6
Query 559: a man and woman inside a car
1
2
3
KELLYTV.BLIP.TV
4
5
6

Figure 3. **Top six shots retrieved from the IACC.3 collection (of 335k shots) with respect to four queries in the TRECVID 2017 AVS task**. Dual encoding is used. Videos without green or red marks are unlabeled. For query #559, the second, third and fifth shots are unlabeled but seem to be relevant to the query. The fourth shot is incorrect, as our video retrieval model fails to recognize that the man is in fact outside a car.

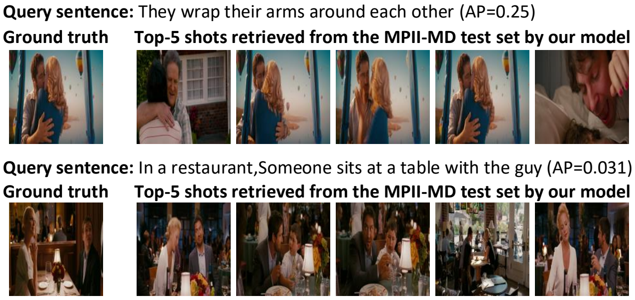
Query sentence: They wrap their arms around each other (AP=0.25)
Ground truth
Top-5 shots retrieved from the MPII-MD test set by our model
Query sentence: In a restaurant,Someone sits at a table with the guy (AP=0.031)
Ground truth
Top-5 shots retrieved from the MPII-MD test set by our model

Figure 4. **Movie retrieval by sentence on MPII-MD**. The top retrieved shots, though not being ground truth, appear to be correct.

side multi-level encoding is also beneficial for VSE++ in its original context.

5.5. Efficiency Test

Recall that the dual encoding network is designed to represent both videos and sentences into a common space respectively. Once the network is trained, representing them in the common space can be performed independently. This means we can process large-scale videos offline and answer ad-hoc queries on the fly. Specifically, given a natural-sentence query, it takes approximately 0.14 second to retrieve videos from the largest IACC.3 dataset, which consists of 335,944 videos. The performance is tested on a normal computer with 64G RAM and a GTX 1080TI GPU. The retrieval speed is adequate for instant response.

6. Summary and Conclusions

For zero-example video retrieval this paper proposes dual encoding. By jointly exploiting multiple encoding strategies at different levels, the proposed dual encoding network encodes both videos and natural language queries into powerful dense representations. Followed by common space learning, these representations can be transformed to perform sequence-to-sequence cross-modal matching effectively. Extensive experiments on three benchmarks, *i.e.* MSR-VTT, TRECVID 2016 and 2017 AVS tasks, support the following conclusions. Among the three levels of encoding, biGRU-CNN that builds a 1-d convolutional network on top of bidirectional GRU is the most effective when used alone. Video-side multi-level encoding is more beneficial when compared with its text-side counterpart. For state-of-the-art performance, we recommend dual encoding. We believe the proposed method also has a potential for other tasks such as video question answering that require effective video / text encoding.

Acknowledgments

This work was supported by NSFC (No. 61672523, No. 61773385, No. U1609215, No. 61772466), ZJNSF (No. LQ19F020002), the Fundamental Research Funds for the Central Universities and the Research Funds of Renmin University of China (No. 18XNLG19), and the Zhejiang Provincial Natural Science Foundation for Distinguished Young Scholars (No. LR19F020003).

References

[1] G. Awad, A. Butt, J. Fiscus, D. Joy, A. Delgado, M. Michel, A. F. Smeaton, Y. Graham, W. Kraaij, G. Qunot, M. Eskevich, R. Ordelman, G. J. F. Jones, and B. Huet. TRECVID 2017: Evaluating ad-hoc and instance video search, events detection, video captioning and hyperlinking. In *TRECVID Workshop*, 2017.

[2] 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 Workshop*, 2016.

[3] X. Chang, Y. Yang, A. G. Hauptmann, E. P. Xing, and Y.-L. Yu. Semantic concept discovery for large-scale zero-shot event detection. In *AAAI*, 2015.

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

[5] 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.

[6] J. Dalton, J. Allan, and P. Mirajkar. Zero-shot video retrieval using content and concepts. In *CIKM*, 2013.

[7] J. Dong, X. Li, and C. G.M. Snoek. Predicting visual features from text for image and video caption retrieval. *IEEE Trans. Multimedia*, 20(12):3377–3388, 2018.

[8] F. Faghri, D. J Fleet, J. R. Kiros, and S. Fidler. VSE++: Improved visual-semantic embeddings. In *BMVC*, 2018.

[9] A. Habibian, T. Mensink, and C. G. M. Snoek. Composite concept discovery for zero-shot video event detection. In *ICMR*, 2014.

[10] A. Habibian, T. Mensink, and C. G. M. Snoek. Video2vec embeddings recognize events when examples are scarce. *T-PAMI*, 39(10):2089–2103, 2017.

[11] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[12] L. Jiang, D. Meng, T. Mitamura, and A. G. Hauptmann. Easy samples first: Self-paced reranking for zero-example multimedia search. In *ACMMM*, 2014.

[13] Y. Kim. Convolutional neural networks for sentence classification. In *EMNLP*, 2014.

[14] D. P Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[15] D. Le, S. Phan, V.-T. Nguyen, B. Renoust, T. A. Nguyen, V.-N. Hoang, T. D. Ngo, M.-T. Tran, Y. Watanabe, M. Klinkigt, et al. NII-HITACHI-UIT at TRECVID 2016. In *TRECVID Workshop*, 2016.

[16] X. Li, J. Dong, C. Xu, J. Cao, X. Wang, and G. Yang. Renmin University of China and Zhejiang Gongshang University at TRECVID 2018: Deep cross-modal embeddings for video-text retrieval. In *TRECVID Workshop*, 2018.

[17] Y. Li, Y. Song, L. Cao, J. Tetreault, L. Goldberg, A. Jaimes, and J. Luo. TGIF: A new dataset and benchmark on animated GIF description. In *CVPR*, 2016.

[18] J. Liang, J. Chen, P. Huang, X. Li, L. Jiang, Z. Lan, P. Pan, H. Fan, Q. Jin, J. Sun, et al. Informedia @ Trecvid 2016. In *TRECVID Workshop*, 2016.

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

[20] Y.-J. Lu, H. Zhang, M. de Boer, and C.-W. Ngo. Event detection with zero example: Select the right and suppress the wrong concepts. In *ICMR*, 2016.

[21] F. Markatopoulou, D. Galanopoulos, V. Mezaris, and I. Patras. Query and keyframe representations for ad-hoc video search. In *ICMR*, 2017.

[22] F. Markatopoulou, A. Moumtzidou, D. Galanopoulos, T. Mironidis, V. Kaltsa, A. Ioannidou, S. Symeonidis, K. Avgerinakis, S. Andreadis, et al. ITI-CERTH participation in TRECVID 2016. In *TRECVID Workshop*, 2016.

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

[24] N. C. Mithun, J. Li, F. Metze, and A. K. Roy-Chowdhury. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In *ICMR*, 2018.

[25] P. A. Nguyen, Q. Li, Z.-Q. Cheng, Y.-J. Lu, H. Zhang, X. Wu, and C.-W. Ngo. Vireo @ TRECVID 2017: Video-to-text, ad-hoc video search and video hyperlinking. In *TRECVID Workshop*, 2017.

[26] A. Rohrbach, Marcus R., N. Tandon, and B. Schiele. A dataset for movie description. In *CVPR*, 2015.

[27] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Trans. Signal Processing*, 45(11):2673–2681, 1997.

[28] C. G. M. Snoek, X. Li, C. Xu, and D. C. Koelma. University of Amsterdam and Renmin university at TRECVID 2017: Searching video, detecting events and describing video. In *TRECVID Workshop*, 2017.

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

[30] K. Ueki, K. Hirakawa, K. Kikuchi, T. Ogawa, and T. Kobayashi. Waseda_Meisei at TRECVID 2017: Ad-hoc video search. In *TRECVID Workshop*, 2017.

[31] S. Wu, S. Bondugula, F. Luisier, X. Zhuang, and P. Natarajan. Zero-shot event detection using multi-modal fusion of weakly supervised concepts. In *CVPR*, 2014.

[32] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

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

[34] 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.

[35] 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*, 2:67–78, 2014.

[36] Y. Yu, J. Kim, and G. Kim. A joint sequence fusion model for video question answering and retrieval. In *ECCV*, 2018.

[37] 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.

[38] B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus. Simple baseline for visual question answering. *arXiv preprint arXiv:1512.02167*, 2015.