W2VV++: Fully Deep Learning for Ad-hoc Video Search

Xirong Li[1,2,3], Chaoxi Xu[1,2], Gang Yang[2], Zhineng Chen[4], Jianfeng Dong[5]
[1]Key Lab of Data Engineering and Knowledge Engineering, Renmin University of China
[2]AI & Media Computing Lab, School of Information, Renmin University of China
[3]School of Information Science and Technology, University of Science and Technology of China
[4]Interactive Media Research Center, Institute of Automation, Chinese Academy of Sciences
[5]College of Computer and Information Engineering, Zhejiang Gongshang University

ABSTRACT

Ad-hoc video search (AVS) is an important yet challenging problem in multimedia retrieval. Different from previous concept-based methods, we propose a fully deep learning method for query representation learning. The proposed method requires no explicit concept modeling, matching and selection. The backbone of our method is the proposed *W2VV++* model, a super version of Word2VisualVec (W2VV) previously developed for visual-to-text matching. W2VV++ is obtained by tweaking W2VV with a better sentence encoding strategy and an improved triplet ranking loss. With these simple yet important changes, W2VV++ brings in a substantial improvement. As our participation in the TRECVID 2018 AVS task and retrospective experiments on the TRECVID 2016 and 2017 data show, our best single model, with an overall inferred average precision (infAP) of 0.157, outperforms the state-of-the-art. The performance can be further boosted by model ensemble using late average fusion, reaching a higher infAP of 0.163. With W2VV++, we establish a new baseline for ad-hoc video search.

CCS CONCEPTS

• **Information systems** → **Video search**; *Query representation.*

KEYWORDS

Ad-hoc video search, query representation learning, cross-modal matching, deep learning, TRECVID benchmarks

ACM Reference Format:
Xirong Li, Chaoxi Xu, Gang Yang, Zhineng Chen, Jianfeng Dong. 2019. W2VV++: Fully Deep Learning for Ad-hoc Video Search. In *Proceedings of the 27th ACM International Conference on Multimedia (MM'19), Oct. 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3343031.3350906

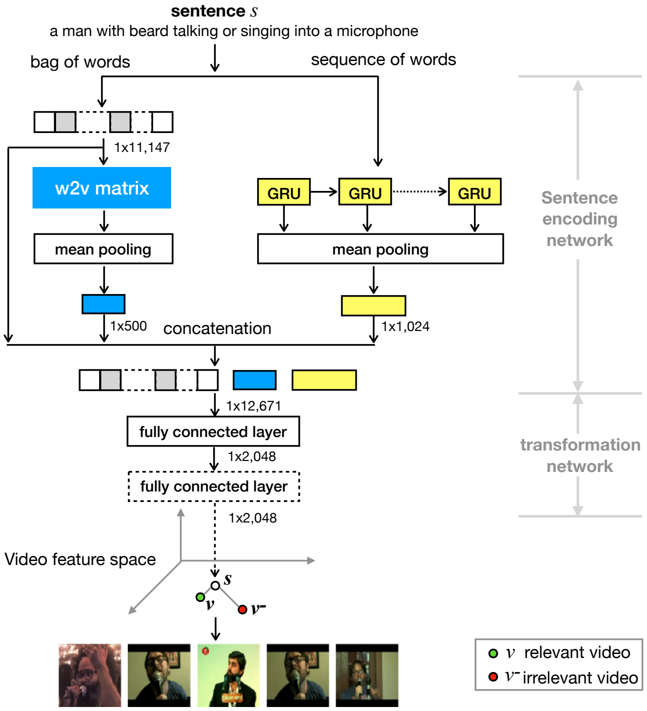

Figure 1: Proposed W2VV++ model for ad-hoc video search. The new model adapts the network architecture of W2VV [8] by exploiting all GRU output vectors by mean pooling. In order to exploit abundant irrelevant sentence-video pairs, W2VV++ is trained to minimize an improved triplet ranking loss [11] instead of the W2VV's mean square error loss which considers only relevant sentence-video pairs.

1 INTRODUCTION

This paper targets at ad-hoc video search (AVS), a challenging problem in multimedia retrieval. An AVS system shall search for *unlabeled* videos relevant with respect to an ad-hoc query expressed exclusively in terms of a natural-language sentence, *e.g., a man with beard talking or singing into a microphone* [3]. The problem thus differs from classical content-based video search [38], where a query is accompanied by image or video examples. The semantic relevance between a textual query and a given video has to be effectively measured in a *cross-modal* manner. AVS also differs from long studied concept-based video retrieval [34], which aims for detecting a specific concept, *i.e.,* an objective linguistic description of an observable entity, from videos. AVS goes beyond this, requiring to model *interconnections between concepts* in the query. We develop an AVS system, see Fig. 1, that meets these two requirements.

The progress in ad-hoc video search has been measured, thanks to good evaluation efforts carried out by the NIST TRECVID AVS

MM'19, October 21–25, 2019, Nice, France

ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3350906

benchmark [1–3]. According to the benchmark evaluation, the majority of the top-performed solutions are concept-based, that is, representing both queries and videos by concept vectors [18, 25, 28, 35], see Table 1. For query representation, a concept vector is constructed by selecting one or few concepts that have the best lexical [23, 35] or semantic [24, 33] match with the query text. For video representation, convolutional neural networks (CNNs) pretrained on visual recognition datasets such as ImageNet [6], EventNet [37], and FC-VID [15] are employed, and the top-predicted concepts are preserved to form a concept vector. While being interpretable, the concept-based representation has two downsides. First, it is difficult to select a right number of "right" concepts that can be reliably detected from the video content and in the meanwhile, informative to describe an ad-hoc query [23]. Second, the importance of a specific concept for query representation is estimated empirically, *e.g.,* in terms of its semantic relatedness to the query [24], and mostly unoptimized for cross-modal similarity computation.

In this paper we investigate a deep learning based alternative that does not require explicit concept modeling, matching and selection. In the context of image retrieval by text, the VSE++ model by Faghri *et al.* [11] uses a gated recurrent unit (GRU) network to model query text. For visual-to-text matching, the Word2VisualVec (W2VV) model by Dong *et al.* [8] uses multiple text encoding strategies, which is found to be more effective than using GRU alone. More recently, Mithun *et al.* [27] adapt VSE++ for video-text retrieval. To the best of our knowledge, the viability of deep learning based query representation has not been proved in large-scale AVS benchmarks such as the TRECVID series.

Our main contributions are two-fold:

- Technically, by improving and repurposing W2VV (originally for image-to-text matching) for AVS, we provide a new means, termed W2VV++, to effectively answer ad-hoc queries described by natural-language text. By contrast, the previous best practices, relying on intensive visual concept modeling, essentially reduce query representation to heuristic keyword matching. Despite its apparent simplicity, W2VV++ is much more effective for ad-hoc video search.
- Conceptually, by winning the TRECVID'18 AVS task, we successfully challenge the concept-based conventions. In fact, our fully automatic run even outperforms the best run from the manually-assisted track where a human formulates the initial query based on topic and query interface (0.121 versus 0.106 in terms of infAP). This paper, as a proof-of-concept, shows for the first time the feasibility of fully deep learning for ad-hoc video search at a large-scale. The finding is further confirmed by our retrospective study on the TRECVID 2016 and 2017 AVS tasks.

2 THE STATE-OF-THE-ART

We review the state-of-the-art in the context of the TRECVID AVS task [3]. We focus on TRECVID as it is the most challenging benchmark, attracting key players in the field [18, 20, 25, 28, 32, 35]. The ground truth was unavailable to participants by the time of the benchmark evaluation. In the meanwhile, participants were requested to not tune their systems for any test query. Such a setup allows one to fairly evaluate how well their models generalize.

Table 1 provides an overview of top-performed systems in the fully automatic track in 2016–2018. We highlight three key designs in these systems, including how a natural-language query is represented (query representation), how an unlabeled video is represented (video representation), and lastly in what feature space cross-modal matching is performed (common space).

In their winning system at TRECVID 2016, Le *et al.* [18] develop a text-based solution. Each video is automatically annotated with visual concepts extracted from video frames by pre-trained CNNs. A standard TF-IDF scheme is used to index the concept-based annotations. The similarity between a given query and a specific video is computed by matching the query text and the video annotations in a textual space. Markatopoulou *et al.* [25] take a similar approach, but develop a rule-based method for concept selection. A semantic-relatedness score [12] between the query and a visual concept is computed. If the score is higher than a given threshold, the concept is selected and used to represent the query. Otherwise a multi-step linguistic analysis [24] is performed to select concepts that partially match the query. Different from [18, 25], Liang *et al.* [20] utilize a webly-labeled learning algorithm [21] to learn a one-versus-all model per query. To tackle the zero-example problem, the authors collect weakly labeled training videos by submitting a given query to YouTube. Although learning individual concepts from YouTube videos is found to be promising [17], it remains challenging to automatically collect a number of relevant videos for complex queries. Indeed, according to the TRECVID 2016 evaluation, the system is found to be less effective than the first two systems.

As for TRECVID 2017, while concept-based query / vector representation remains popular [28, 35], the winning solution by Snoek *et al.* [33] employs a more elegant representation technique called VideoStory [13]. For each unlabeled video, its CNN feature is transformed into a so-called VideoStory embedding by a linear transformation. The embedding is then transformed to a bag-of-words vector by another linear transformation. A query, after heuristically selecting terms based on part-of-the-speech tagging of the query, is also converted to a bag-of-words vector. Consequently, the video-query similarity is implemented as the cosine similarity between their bag-of-words vectors. Despite its good performance, the VideoStory-based solution has two shortcomings. First, bag-of-words ignores sequential information in a query sentence. Second, the effectiveness of bag-of-words counts on proper term selection, which is however disconnected from the representation learning process. In contrast, our solution represents a query sentence by deep neural networks that consider both the importance of query terms and their orders, and are trained end-to-end.

In TRECVID 2018, multiple deep learning based methods for query representation have been tried. The runner-up solution by Huang *et al.* [14] uses two attention networks, besides the classical concept-based representation. Bastan *et al.* [4] experiment with VSE++, reaching the 3rd place in the benchmark evaluation. Our W2VV++ based method performs the best.

While this work focuses on automated search, it can also play an important role in interactive search. As noted by Lokoč *et al.* [22], a common feature of successful interactive search tools in the Video Browser Showdown (VBS) competition is effective query initialization. Our model is likely to be beneficial for the state-of-the-art interactive system [30] that still uses concept detectors.

Table 1: An overview of top-performed systems in the TRECVID 2016 / 2017 / 2018 ad-hoc video search benchmarks (fully automatic track). We describe these systems along three dimensions, namely 1) how an ad-hoc query is represented, 2) how an unlabeled video is represented, and 3) what the cross-modal common space is. Our W2VV++ based solution is the first winning entry that learns to represent ad-hoc queries by deep neural networks in an end-to-end fashion.

Rank	System	Query Representation	Video Representation	Common Space
2016:				
1	Le *et al.* [18]	Bag-of-words	Concept vector extracted by pre-trained CNNs (VGG-16)	A textual space
2	Markatopoulou *et al.* [25]	Concept vector extracted by rule-based concept selection [24]	Concept vector extracted by pre-trained CNNs (AlexNet, GoogLeNet, ResNet, VGGNet)	A 1,345-dim concept space
3	Liang *et al.* [20]	One-versus-all query modeling [21]	Visual features extracted by pre-trained CNNs (VGG-19, C3D)	A visual feature space
2017:				
1	Snoek *et al.* [33]	Bag-of-words	Bag-of-words generated by VideoStory [13]	A textual space
2	Ueki *et al.* [35]	Concept vector extracted by rule-based concept selection	Concept vector extracted by pre-trained CNNs (AlexNet, GoogLeNet)	A 50k-dim concept space
3	Nguyen *et al.* [28]	Concept vector extracted by rule-based concept selection [23]	Concept vector extracted by pre-trained CNNs (ResNet-50)	A 2,774-dim concept space
2018:				
1	*This paper*	Dense vector by W2VV++	Visual features extracted by pre-trained CNNs (ResNeXt-101, ResNet-152)	A visual feature space or learned subspace
2	Huang *et al.* [14]	+ Concept vector extracted by rule-based concept selection + Dense vector by attention networks	+ Concept vector extracted by pre-trained CNNs + Visual features extracted by pre-trained CNNs (ResNet-152)	+ A concept space + A learned subspace
3	Bastan *et al.* [4]	Dense vector by VSE++ [11]	Visual features extracted by pre-trained CNNs (ResNet-152)	A learned subspace

3 OUR METHOD

3.1 Problem Statement

Given an ad-hoc query expressed by a natural-language sentence s of l words $\{w_1, w_2, \dots, w_l\}$, we aim to build a video search system that retrieves videos relevant with respect to the query from a collection of n *unlabeled* videos $\{v_1, v_2, \dots, v_n\}$. The key problem is to construct a cross-modal similarity function $f(s, v) \in \mathcal{R}$ such that the similarity score of a relevant sentence-video pair (s, v^+) will be larger than the similarity score of an irrelevant sentence-video pair (s, v^-). Accordingly, the relevant video v^+ will be ranked ahead of the irrelevant v^- in search results. Let $\mathbf{s}$ and $\mathbf{v}$ be vectorized representations of the query and the video in a common space, respectively. The cross-modal similarity is obtained based on the cosine similarity:

$$f(s, v) := \frac{\mathbf{s}^T \mathbf{v}}{||\mathbf{s}|| \cdot ||\mathbf{v}||}. \tag{1}$$

We focus on query representation learning that predicts $\mathbf{s}$ from the query. Meanwhile, $\mathbf{v}$ can be instantiated using either deep CNN features or concept vectors as exploited in previous works.

3.2 Query Representation Learning

Our model is built on top of the W2VV model [8], originally proposed for image and video caption retrieval. Conceptually, W2VV is composed of two subnetworks, *i.e.,* a sentence encoding network that quantizes a given sentence into a real-value feature vector and a transformation network that projects the feature vector into the video feature space. We improve W2VV with a better sentence encoding strategy and a better loss for model training, and thus we term the new model W2VV++.

The sentence encoding network. Given a sentence s, W2VV++ quantizes s at multiple scales by running three encoding components in parallel, *i.e.,* bag-of-words (bow), word2vec embedding (w2v), and RNN-based sequential modeling. More specifically, the bow encoding is performed as

$$\mathbf{bow}(s) := (c(w_1, s), c(w_2, s), \dots, c(w_m, s)), \tag{2}$$

where $c(w, s)$ counts the occurrence of a specific word w in s, with m as the size of a given vocabulary. The vocabulary used in this work is constructed by first excluding words that occur less than five times in our training dataset, resulting in a set of 11,282 unique words. We then remove those that can be found in the NLTK stopword list, obtaining eventually $m = 11,147$ words.

Given a pretrained w2v model where $\mathbf{e}(w)$ indicates the semantic embedding vector of a specific word w, the w2v based encoding of the sentence is obtained by mean pooling over its words, namely

$$\mathbf{w2v}(s) := \frac{1}{l} \sum_{i=1}^{l} \mathbf{e}(w_i). \tag{3}$$

We adopt a 500-dimensional w2v model [8], trained on English tags associated with 30 million Flickr images. The model supports over 1.7 million words.

Similar to [8] we instantiate the RNN component with Gated recurrent units (GRUs) [5]. The input vector of a GRU at a specific time step t is a word embedding vector of the t-th word in the sentence. Let $\mathbf{e}(w_t)$ be such a vector, obtained by a table lookup from a word embedding matrix W_e. The output vector of the GRU, denoted by h_t, is updated by jointly exploiting $\mathbf{e}(w_t)$ and the previous output vector h_{t-1} as follows:

$$\begin{aligned} z_t &= \sigma_g(W_z\mathbf{e}(w_t) + U_z h_{t-1} + b_z), \\ r_t &= \sigma_g(W_r\mathbf{e}(w_t) + U_r h_{t-1} + b_r), \\ \widetilde{h}_t &= \sigma_h(W_h\mathbf{e}(w_t) + U_h(r_t \circ h_{t-1}) + b_h), \\ h_t &= (1 - z_t) \circ h_{t-1} + z_t \circ \widetilde{h}_t, \end{aligned} \tag{4}$$

where z_t and r_t indicate the update and reset gate vectors at time t, W, U and b with specific subscripts parameterize affine transformations in the corresponding gates. The output of each gate is followed by a specific activation function, with σ_g indicating a sigmoid function and σ_h for a hyperbolic tangent. The operator $\circ$ means the Hadamard product between two vectors.

For GRU-based sentence encoding, *different from W2VV* which uses only the output vector at the last time step, namely h_l, we consider the outputs from all intermediate steps as well, obtaining our GRU-based encoding by mean pooling:

$$\mathbf{gru}(s) = \frac{1}{l}\sum_{i=1}^{l} h_i. \tag{5}$$

The GRU vocabulary is the same as the bow vocabulary except that all stopwords are now preserved as they contain meaningful contextual information in a natural-language sentence.

The multi-scale sentence encoding is obtained by vector concatenation, *i.e.,* $\mathbf{ms}(s) = [\mathbf{bow}(s); \mathbf{w2v}(s); \mathbf{gru}(s)]$. In this work the size of the GRU output vector is set to 1,024. Therefore, the dimensionality of $\mathbf{ms}(s)$ is $11,147 + 500 + 1,024 = 12,671$.

The transformation network. This network is used to transform the output of the previous network, *i.e.,* $\mathbf{ms}(s)$, to $\mathbf{s}$ so that Eq. 1 can be computed. To that end, we utilize a network of k fully connected (FC) layers. The output vector of the first hidden layer, $\mathbf{fc}_1(s)$, is obtained by affine transformation on $\mathbf{ms}(s)$, *i.e.,*

$$\mathbf{fc}_1(s) = \sigma(A_1\mathbf{ms}(s) + b_1), \tag{6}$$

where A_1 and b_1 are layer weights and bias, while σ is the activation function for increasing the nonlinearity of the network, which is ReLU unless otherwise stated. The output vector of a subsequent layer is computed as

$$\mathbf{fc}_i(s) = \sigma(A_i\mathbf{fc}_{i-1}(s) + b_i),\ i = 2, \ldots, k. \tag{7}$$

The output vector of the last hidden layer is used to represent the query, *i.e.,* $\mathbf{s} = \mathbf{fc}_k(s)$.

Note that the sentence encoding network and the transform network can be jointly trained in an end-to-end manner. To make this notion more explicit, we put all the learnable parameters $\{W_z, U_z, b_z, W_r, U_r, b_r, W_h, U_h, b_h, W_e, A_1, b_1, \ldots, A_k, b_k\}$ together as θ. Accordingly, the similarity function is parameterized as $f(s, v; \theta)$.

Loss function. *Different from W2VV* trained to minimize the mean square error loss, we adopt an improved triplet ranking loss [11], which achieves the state-of-the-art in the image domain. While the classical triplet ranking loss selects the negative example by random, the improved loss selects the hardest negative that violates the ranking constraint the most. More specifically, the loss in the current context is defined as

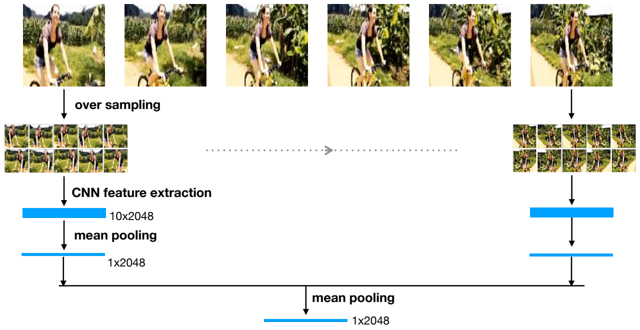

Figure 2: Video representation, obtained by first using a pre-trained image CNN to extract frame-level features in an over-sampling manner and then mean pooling.

$$loss(s;\theta) = \max_{v^-}(0, \alpha + f(s, v^-;\theta) - f(s, v^+;\theta)), \tag{8}$$

where α is a nonnegative hyper parameter controlling the margin. The hardest negative example is practically selected from each mini-batch during training.

3.3 Video Representation

As aforementioned, this work targets at query representation learning. So for video representation, we simply use deep visual features extracted by state-of-the-art CNNs in an over-sampling manner followed by mean pooling, see Fig. 2. For more advanced video representation, we refer to [9]. We employ two CNNs: ResNeXt-101 used in [33] and ResNet-152 used in [7]. For each model, we take the input of the classification layer as the feature, which has a dimensionality of 2,048. For a given video, we uniformly sample frames with an interval of 0.5 second. Each frame is resized to 256 × 256. CNN features are extracted from its 10 sub images, which are generated by clipping the frame and its horizontal flip with a window of 224 × 224 at their center and four corners. The 10 features are averaged as the frame-level feature. Accordingly, two 2,048-dim video-level features are obtained by mean pooling over frames. For the ease of reference, we denote the two features as *ResNeXt* and *Resnet*, respectively, and use *ResNeXt-Resnet* to refer to their concatenation. When predicting $\mathbf{s}$ with respect to ResNeXt or Resnet, the last hidden layer of W2VV++ has a shape of $2,048 \times 1$. When predicting ResNeXt-Resnet, the shape has to be adjusted accordingly to $4,096 \times 1$.

In principle, W2VV++ can be used to predict any video features including concept vectors [23, 24, 26], 3D-CNN features [27] and their combinations. We leave this direction for future investigation.

3.4 W2VV++ Implementation

We implement W2VV++ using the PyTorch framework [29]. To train W2VV++, We employing the RMSProp optimizer, using its default parameters except for the learning rate, which is empirically set to

0.0001. To avoid gradient explosion, we clip gradients by scaling them down by their l_2 norm. The learning rate is decayed by a factor of 0.99 after each epoch. If the performance on a validation set does not increase in three consecutive epochs, the learning rate is divided by 2 [16]. If the performance does not increase in ten consecutive epochs, an early stop is applied. We pick the model that has the best performance on the validation set.

A mini-batch consists of 128 relevant sentence-video pairs. For each sentence in a given batch, we calculate its loss by simply treating videos from the other pairs irrelevant, and thus selecting the hard negative example from these videos. The margin parameter α in Eq. 8 is set to 0.2. To prevent overfitting, dropout with a rate of 0.2 is applied on the hidden layers in the transformation network.

4 EVALUATION

To justify the effectiveness of our W2VV++ based solution, we conduct two sets of experiments. The first set is based on our participation in the TRECVID 2018 AVS task, where each team was allowed to submit four runs at maximum. This restriction limits our investigation. Therefore, in the second set we perform a retrospective study, providing a more comprehensive evaluation using the TRECVID AVS queries from the last three years (2016–2018).

4.1 Experimental Setup

Training / validation / test sets used in our experiments are as follows, with basic statistics summarized in Table 2.

Training set. We combine the MSR-VTT [36] and TGIF [19] datasets as our training set. The MSR-VTT dataset contains 10K web video clips and 200k natural sentences describing the visual content of the clips. The average number of sentences per clip is 20. From each clip we sampled frames uniformly, obtaining 305,462 frames in total. The TGIF dataset contains 100K animated GIFs and 120K sentences describing visual content of the GIFs. We again sampled frames uniformly, obtaining 1,045,268 frames in total.

Validation set. We adopt the training set of the TRECVID 2016 Video-to-Text task [3], termed TV16-VTT-train, for model selection. This set consists of 200 videos, each associated with two sentences. For each video, we use its first sentence as a textual query, simply considering the other 199 videos irrelevant with respect to this query. Accordingly, we use the mean reciprocal rank (MRR) to measure the performance of a specific model on the validation set.

Test set. We test on IACC.3, the official test set for the TRECVID AVS task 2016–2018 [3]. The set contains 4,593 Internet archive videos (600 hours) with Creative Commons licenses in MPEG-4/H.264 format. Video duration ranges from 6.5 minutes to 9.5 minutes, with a mean duration of approximately 7.8 minutes. Automated shot boundary detection has been performed by the task organizers, resulting in 335,944 video clips in total. From each clip we sampled frames uniformly, obtaining 3,845,221 frames in total.

Test queries. Each year the TRECVID AVS task organizers define a list of 30 test queries. Each query is presented in terms of a natural-language sentence with varied length and varied visual and semantic complexity. Examples are "Find shots of palm trees", "Find shots of a man with beard and wearing white robe speaking and gesturing to camera", and "Find shots of a truck standing still while a person is walking beside or in front of it". All the queries start with the phrase "Find shots of", which was removed automatically.

Table 2: Datasets used in this paper. Notice that the training and validation datasets were independently constructed by their developers for other purposes, *i.e.,* MSR-VTT for video captioning [36], TGIF for gif captioning [19], and TV16-VTT-train for video-to-text matching [3].

Dataset	Shots	Frames	Sentences	Words for training
For training:				
MSR-VTT	10,000	305,462	200,000	9,707
TGIF	100,855	1,045,268	124,534	4,959
For validation:				
TV16-VTT-train	200	5,941	200	–
For testing:				
IACC.3	335,944	3,845,221	–	–

Performance metric. We report inferred average precision (infAP), the official performance metric [1–3]. The overall performance of a video search system is measured by averaging infAP scores of all test queries.

4.2 Experiments

4.2.1 Experiment 1. TRECVID 2018 Participation. We participated in the TRECVID 2018 AVS task with the following four models,

(1) *W2VV++ (ResNeXt-Resnet)*: A W2VV++ model that predicts the combined feature of ResNeXt-101 and ResNet-152 for a given sentence.
(2) *W2VV++ (ResNeXt, subspace)*: This model differs from the previous one in two aspects. First, it predicts only the ResNeXt feature. Second, it adds a FC layer on top of the video feature for feature re-learning.
(3) *W2VV++ (ResNeXt-Resnet, subspace)*: Similar to the second model, but predicting the combined feature.
(4) *Ensemble*: Late average fusion of dozens of W2VV++ models, which are trained with varied setups including the choice of video features (ResNeXt, Resnet or ResNeXt-Resnet), the choice of activation functions (ReLU or hyperbolic tangent) and the number of hidden layers ($k = 1, 2$) in the transformation network, and the dimensionality of the word embedding for GRUs (300 or 500).

An overview of the TRECVID 2018 benchmark results is shown in Fig. 3. *W2VV++ (ResNeXt-Resnet)*, serving as our baseline, is better than all submissions from the other teams. The result clearly shows the effectiveness of our W2VV++ based video search system. *W2VV++ (ResNeXt-Resnet, subspace)*, by adding an additional feature re-learning layer, outperforms *W2VV++ (ResNeXt-Resnet)*, though the improvement appears to be marginal. The Ensemble model gives a noticeable performance boost, suggesting that the meta models are complementary to each other.

4.2.2 Experiment 2. Retrospective Study . To better understand the influence of the video features, we train two more W2VV++ models, one for predicting ResNeXt and the other for predicting Resnet,

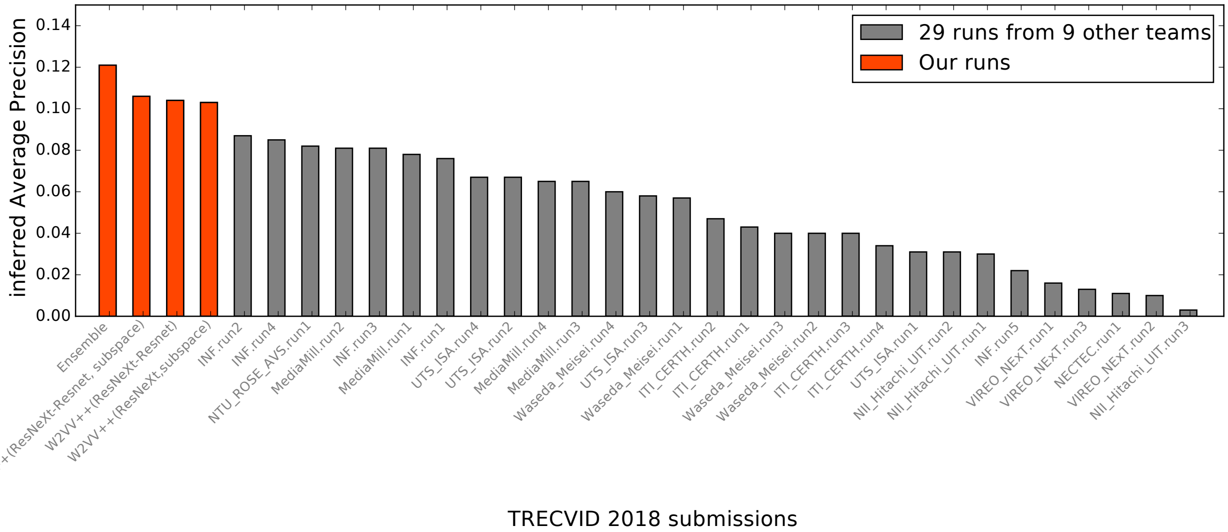

Figure 3: Overview of the TRECVID 2018 AVS task benchmark, all runs ranked according to mean infAP.

denoted as *W2VV++ (ResNeXt)* and *W2VV++ (Resnet)*, respectively. In the remaining part of the experiments, we use *W2VV++ (ResNeXt)* as a reference for comparison.

Comparison to the state-of-the-art. Since our retrospective study covers all the test queries from the last three years, we compare with the following:

- Top-3 systems at TRECVID 2016, *i.e.,* [18], [25] and [20], and at TRECVID 2017, *i.e.,* [33], [35] and [28].
- Markatopoulou *et al.* [24], as an improved version of [25].
- VideoStory [13], the best single model that contributes to the winning solution of [33] at TRECVID 2017.
- W2VV [8], upon which W2VV++ is developed.
- VSE++ [11], the loss of which is used by W2VV++.

The performance is summarized in Table 3. Note that the scores of the existing methods are directly cited from their papers or technical reports except for VSE++ and W2VV. These two works have released their source codes, allowing us to re-train their models with our experimental setup. Our four models from Experiment 1 (corresponding to the last four rows in Table 3), used as is, outperform the previous best runs. It is worth noting the top-1 result by [33], with an infAP of 0.206, is obtained by combining a number of meta models. It is surpassed by *W2VV++ (ResNeXt-Resnet, subspace)*, a single model reaching a higher infAP of 0.213. These results again confirm the effectiveness of W2VV++.

Note the video pool stays the same while the queries change each year. Our retrospective experiment suggests that the 2018 queries are the most difficult, while the 2017 queries seem to be the easiest.

Table 3: Retrospective experiments on the TRECVID 2016 / 2017 / 2018 AVS tasks. For W2VV++ with varied setups, we use W2VV++ (ResNeXt) as a reference. Relative improvements over this reference are shown in parentheses.

	TRECVID edition			
	2016	*2017*	*2018*	**OVERALL**
Top-3 TRECVID finalists:				
Rank 1	0.054 [18]	0.206 [33]	0.121	–
Rank 2	0.051 [25]	0.159 [35]	0.087 [14]	–
Rank 3	0.040 [20]	0.120 [28]	0.082 [4]	–
Literature methods:				
VSE++ (ResNeXt) [11]	0.123	0.154	0.074	0.117 (↓ -10.7%)
VideoStory [13]	0.087	0.150	–	–
Markatopoulou *et al.* [24]	0.064	–	–	–
W2VV (ResNeXt) [8]	0.050	0.081	0.013	0.048 (↓ -63.4%)
Our W2VV++:				
ResNeXt	0.137	0.168	0.088	0.131
Resnet	0.126	0.151	0.089	0.122 (↓ -6.9%)
ResNeXt-Resnet	0.149	0.176	0.104	0.143 (↑ 9.2%)
ResNeXt, subspace	0.140	0.171	0.103	0.138 (↑ 5.3%)
ResNeXt-Resnet, subspace	**0.151**	0.213	0.106	0.157 (↑ 19.8%)
Ensemble	0.149	**0.220**	**0.121**	**0.163** (↑ 24.4%)

Test of statistical significance. We perform a randomization test [31], which is valid if permutations of retrieval system A and B are fully random. This assumption is fulfilled with ease by the test protocol, and thus applicable to most IR experiments [10, 31]. The test result is listed in Table 4. Except for *W2VV++ (ResNeXt, subspace)*, the performance difference between *W2VV++ (ResNeXt)* and the other models passes the randomization test at the significance level of $p = 0.05$. We interpret the result as follows. Towards building a more effective W2VV++ model, a better video feature is preferred to video feature re-learning. Nonetheless, the two can be exploited together for even better performance.

Per-query analysis. For a more comprehensive picture of our results, we visualize per-query performance in Fig. 4. After observing the search results, we find that an infAP of over 0.1 often means a number of relevant examples are included in the top-20 results. For nearly 54% of the queries, our best single model *W2VV++*

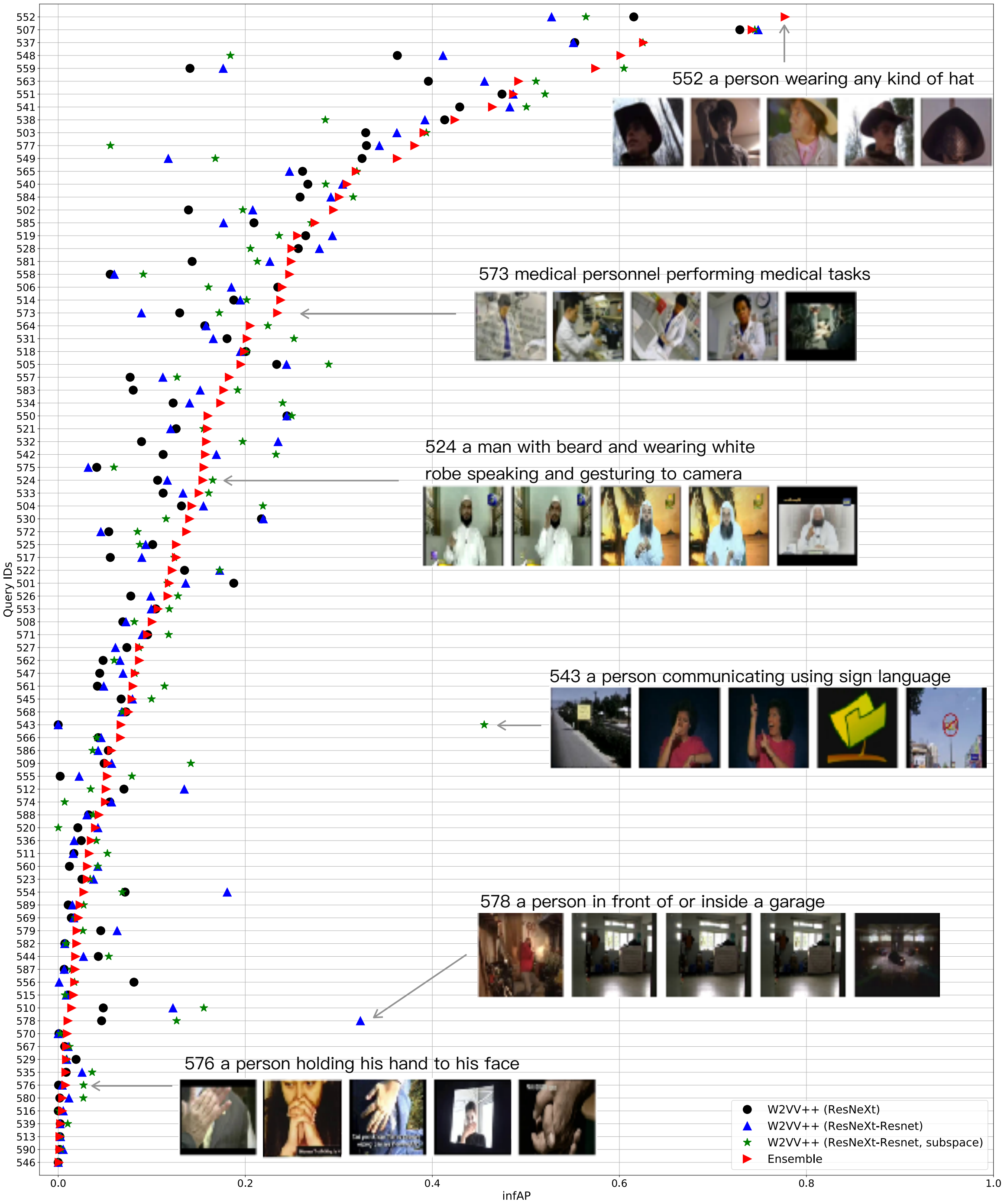

Figure 4: Performance of all the 90 queries from the TRECVID 2016 / 2017 / 2018 AVS tasks. The queries are sorted in descending order according to the performance of the *Ensemble* model. Images below a given query indicate the top-5 videos retrieved by the best model with respect to the given query. For 48 out of the 90 queries, the best single model, *i.e., W2VV++ (ResNeXt-Resnet, subspace),* obtains infAP scores larger than 0.1. Best viewed in color.

(ResNeXt-Resnet, subspace) meets this criterion, making it promising for precision-oriented video search.

We also look into failed queries such as query #576 "*a person holding his hand to his face*". As Fig. 4 shows, the top-5 hits contain either purely hands or a woman holding her hands to her face.

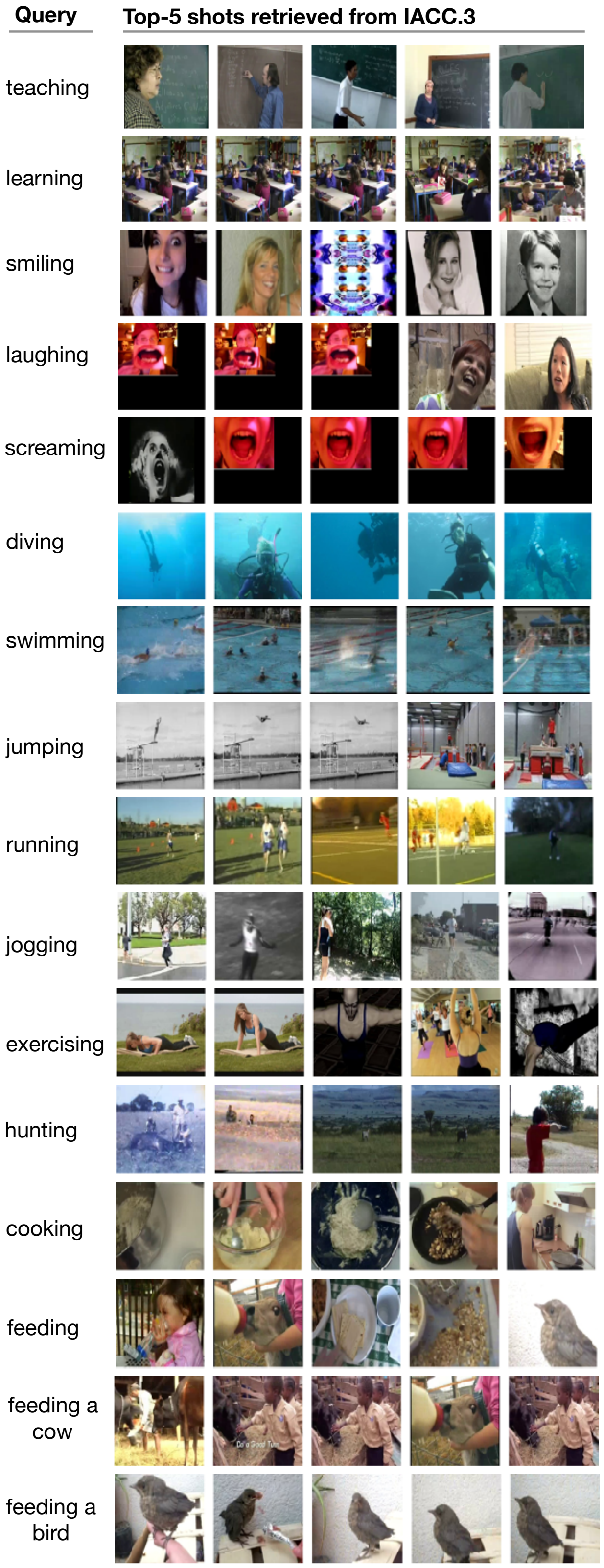

Figure 5: Showcase of answering single-word queries by *W2VV++ (ResNeXt-Resnet)*. Best viewed in color.

Table 4: Test of statistical significance. W2VV++(ResNeXt) is used as a reference. Except for W2VV++(ResNeXt, subspace), the performance difference of the other models against this reference is found to be statistically significant.

Model	p-value	Significant?
W2VV (ResNeXt)	0.0000	✓
VSE++ (ResNeXt)	0.0492	✓
W2VV++ (ResNeXt-Resnet)	0.0138	✓
W2VV++ (ResNeXt, subspace)	0.2083	✗
W2VV++ (ResNeXt-Resnet, subspace)	0.0034	✓
Ensemble	0.0000	✓

Recall that our model is learned from the video descriptions of the MSR-VTT and TGIF datasets, where the joint use of the key terms *holding*, *hand* and *face* is relatively rare. An interesting future work is to automatically paraphrase an ad-hoc query so that it better fits the data where the model is learned from.

Although W2VV++ is intended for representing natural-language queries, it also works with queries of few or one word. Fig. 5 shows how the model responses to daily actions such as *teaching*, *learning*, *smiling*, *laughing*, and *swimming*. In addition to the TRECVID experiments, we compare with Mithun *et al.* [27], SOTA by the time of MM’19 submission on MSR-VTT, using the same ResNet-152 feature and the same data split. W2VV++ is better in terms of R@1 (7.0 vs 5.8), R@5 (20.5 vs 17.6) and R@10 (29.3 vs 25.2).

5 CONCLUSIONS

For ad-hoc video search, we propose W2VV++ that learns to represent an ad-hoc query by deep neural networks. Different from previous concept-based methods, W2VV++ is concept-free. Moreover, the proposed model can be trained in an end-to-end manner, enabling a joint optimization of query representation learning and cross-modality similarity computation. From the experimental results on the TRECVID 2016 / 2017 / 2018 AVS benchmarks, we arrive at the following conclusions:

- For building a more effective W2VV++ model, a better video feature is preferred to video feature re-learning. The two can be exploited together for even better performance.
- We recommend *W2VV++ (ResNeXt-Resnet, subspace)* as the best single model to use. For state-of-the-art results, we recommend the *Ensemble* model.
- Even though W2VV++ is intended for representing natural-language queries, it also works with queries of few or one word.

W2VV++, with room for future improvement, has established a new baseline for ad-hoc video search.

ACKNOWLEDGMENTS

This work was supported by NSFC (No. 61672523, No. 61771468), BJNSF (No. 4192029), ZJNSF (No. LQ19F020002), the Fundamental Research Funds for the Central Universities and the Research Funds of Renmin University of China (No. 18XNLG19). Corresponding author: Jianfeng Dong.

REFERENCES

[1] G. Awad, A. Butt, K. Curtis, Yo. Lee, J. Fiscus, A. Godil, D. Joy, A. Delgado, A. F. Smeaton, Y. Graham, W. Kraaij, G. Quénot, J. Magalhaes, D. Semedo, and S. Blasi. 2018. TRECVID 2018: Benchmarking Video Activity Detection, Video Captioning and Matching, Video Storytelling Linking and Video Search. In *TRECVID*.

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

[3] G. Awad, J. Fiscus, D. Joy, M. Michel, A. Smeaton, W. Kraaij, G. Quénot, M. Eskevich, R. Aly, R. Ordelman, G. Jones, B. Huet, and M. Larson. 2016. TRECVID 2016: Evaluating Video Search, Video Event Detection, Localization, and Hyperlinking. In *TRECVID*.

[4] M. Bastan, X. Shi, J. Gu, Z. Heng, C. Zhuo, D. Sng, and A. Kot. 2018. NTU ROSE Lab at TRECVID 2018: Ad-hoc Video Search and Video to Text. In *TRECVID*.

[5] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. 2014. Learning Phrase Representations using RNN Encoder-decoder for Statistical Machine Translation. In *EMNLP*.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: a Large-scale Hierarchical Image Database. In *CVPR*.

[7] J. Dong, S. Huang, D. Xu, and D. Tao. 2017. DL-61-86 at TRECVID 2017: Video-to-Text Description. In *TRECVID*.

[8] J. Dong, X. Li, and C. G. M. Snoek. 2018. Predicting Visual Features from Text for Image and Video Caption Retrieval. *T-MM* 20, 12 (2018), 3377–3388.

[9] J. Dong, X. Li, C. Xu, S. Ji, Y. He, G. Yang, and X. Wang. 2019. Dual Encoding for Zero-Example Video Retrieval. In *CVPR*.

[10] J. Dong, X. Li, and D. Xu. 2018. Cross-Media Similarity Evaluation for Web Image Retrieval in the Wild. *T-MM* 20, 9 (2018), 2371–2384.

[11] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler. 2018. VSE++: Improving Visual-Semantic Embeddings with Hard Negatives. In *BMVC*.

[12] E. Gabrilovich and S. Markovitch. 2007. Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In *IJCAI*.

[13] A. Habibian, T. Mensink, and C. G. M. Snoek. 2017. Video2vec Embeddings Recognize Events When Examples Are Scarce. *T-PAMI* 39, 10 (2017), 2089–2103.

[14] P.-Y. Huang, J. Liang, V. Vaibhav, X. Chang, and A. Hauptmann. 2018. Informedia@TRECVID 2018: Ad-hoc Video Search with Discrete and Continuous Representations. In *TRECVID*.

[15] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, and S.-F. Chang. 2018. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *T-PAMI* 40, 2 (2018), 352–364.

[16] A. Joulin, L. van der Maaten, A. Jabri, and N. Vasilache. 2016. Learning Visual Features from Large Weakly Supervised Data. In *ECCV*.

[17] S. Kordumova, X. Li, and C. Snoek. 2015. Best practices for learning video concept detectors from social media examples. *MTAP* 74, 4 (2015), 1291–1315.

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

[19] Y. Li, Y. Song, L. Cao, J. Tetreault, L. Goldberg, A. Jaimes, and J. Luo. 2016. TGIF: A New Dataset and Benchmark on Animated GIF Description. In *CVPR*.

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

[21] J. Liang, L. Jiang, D. Meng, and A. Hauptmann. 2016. Learning to Detect Concepts from Webly-labeled Video Data. In *IJCAI*.

[22] J. Lokoč, W. Bailer, K. Schoeffmann, B. Muenzer, and G. Awad. 2018. On Influential Trends in Interactive Video Retrieval: Video Browser Showdown 2015–2017. *T-MM* 20, 12 (2018), 3361–3376.

[23] Y.-J. Lu, H. Zhang, M. de Boer, and C.-W. Ngo. 2016. Event Detection with Zero Example: Select the Right and Suppress the Wrong Concepts. In *ICMR*.

[24] F. Markatopoulou, D. Galanopoulos, V. Mezaris, and I. Patras. 2017. Query and Keyframe Representations for Ad-hoc Video Search. In *ICMR*.

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

[26] M. Merler, B. Huang, L. Xie, G. Hua, and A. Natsev. 2012. Semantic Model Vectors for Complex Video Event Recognition. *T-MM* 14, 1 (2012).

[27] N. Mithun, J. Li, F. Metze, and A. K. Roy-Chowdhury. 2018. Learning Joint Embedding with Multimodal Cues for Cross-Modal Video-Text Retrieval. In *ICMR*.

[28] P. Nguyen, Q. Li, Z.-Q. Cheng, Y.-J. Lu, H. Zhang, X. Wu, and C.-W. Ngo. 2017. VIREO @ TRECVID 2017: Video-to-Text, Ad-hoc Video Search and Video Hyperlinking. In *TRECVID*.

[29] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. 2017. Automatic differentiation in PyTorch. In *NIPS-W*.

[30] L. Rossetto, M. Amiri Parian, R. Gasser, I. Giangreco, S. Heller, and H. Schuldt. 2019. Deep Learning-Based Concept Detection in vitrivr. In *MMM*.

[31] M. Smucker, J. Allan, and B. Carterette. 2007. A Comparison of Statistical Significance Tests for Information Retrieval Evaluation. In *CIKM*.

[32] C. G. M. Snoek, J. Dong, X. Li, X. Wang, Q. Wei, W. Lan, E. Gavves, N. Hussein, D. C. Koelma, and A. W. M. Smeulders. 2016. University of Amsterdam and Renmin University at TRECVID 2016: Searching Video, Detecting Events and Describing Video. In *TRECVID*.

[33] C. G. M. Snoek, X. Li, C. Xu, and D. C. Koelma. 2017. University of Amsterdam and Renmin University at TRECVID 2017: Searching Video, Detecting Events and Describing Video. In *TRECVID*.

[34] C. G. M. Snoek and M. Worring. 2009. Concept-Based Video Retrieval. *Found. Trends Inf. Retr.* 2, 4 (2009), 215–322.

[35] K. Ueki, K. Hirakawa, K. Kikuchi, T. Ogawa, and T. Kobayashi. 2017. Waseda_Meisei at TRECVID 2017: Ad-hoc Video Search. In *TRECVID*.

[36] J. Xu, T. Mei, T. Yao, and Y. Rui. 2016. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In *CVPR*.

[37] G. Ye, Y. Li, H. Xu, D. Liu, and S.-F. Chang. 2015. EventNet: A Large Scale Structured Concept Library for Complex Event Detection in Video. In *ACMMM*.

[38] S.-I. Yu, L. Jiang, Z. Xu, Y. Yang, and A. G. Hauptmann. 2015. Content-Based Video Search over 1 Million Videos with 1 Core in 1 Second. In *ICMR*.