University of Amsterdam and Renmin University at TRECVID 2016: Searching Video, Detecting Events and Describing Video

Cees G. M. Snoek*, Jianfeng Dong†‡, Xirong Li†, Xiaoxu Wang†, Qijie Wei†, Weiyu Lan†,
Efstratios Gavves*, Noureldien Hussein*, Dennis C. Koelma*, Arnold W. M. Smeulders*

*University of Amsterdam
Amsterdam, The Netherlands

†Renmin University of China
Beijing, China

‡Zhejiang University
Hangzhou, China

Abstract

In this paper we summarize our TRECVID 2016 [1] video recognition experiments. We participated in three tasks: video search, event detection and video description. Here we describe the tasks on event detection and video description. For event detection we explore semantic representations based on VideoStory and an ImageNet Shuffle for both zero-shot and few-example regimes. For the showcase task on video description we experiment with a deep network that predicts a visual representation from a natural language description, and use this space for the sentence matching. For generative description we enhance a neural image captioning model with Early Embedding and Late Reranking. The 2016 edition of the TRECVID benchmark has been a fruitful participation for our joint-team, resulting in the best overall result for zero- and few-example event detection as well as video description by matching and in generative mode.

1 Task I: Event Recognition

The MediaMill approach to multimedia event detection is optimized for recognition scenarios when video examples are scarce or even completely absent. The key in such a challenging setting is a semantic video representation [13]. Our experiments focus on exploring such semantic representations for video search.

1.1 Representation I: VideoStory

The first representation is based on VideoStory, as detailed in [6, 7]. To summarize, it learns the video representation from freely available web videos and their descriptions using an embedding between video features and term vectors. In the embedding the correlations between the words are utilized to learn a more effective representation by optimizing a joint objective balancing descriptiveness and predictability. We start from a dataset of videos, represented by video features $\boldsymbol{X}$, and their textual descriptions, represented by binary term vectors $\boldsymbol{Y}$, indicating which words are present in each video description. Then, our VideoStory representation is learned by minimizing:

$$L_{\mathrm{V}}(\boldsymbol{A},\boldsymbol{W}) = \min_{\boldsymbol{S}} L_d(\boldsymbol{A},\boldsymbol{S}) + L_p(\boldsymbol{S},\boldsymbol{W}), \qquad (1)$$

where $\boldsymbol{A}$ is the textual projection matrix, $\boldsymbol{W}$ is the visual projection matrix, and $\boldsymbol{S}$ is the VideoStory embedding. The loss function L_d corresponds to our first objective for learning a descriptive VideoStory, and the loss function L_p corresponds to our second objective for learning a predictable VideoStory. The embedding $\boldsymbol{S}$ interconnects the two loss functions.

Descriptiveness. For the L_d function, we use a variant of regularized Latent Semantic Indexing. This objective minimizes the quadratic error between the original video descriptions $\boldsymbol{Y}$, and the reconstructed translations obtained from $\boldsymbol{A}$ and $\boldsymbol{S}$:

$$L_d(\boldsymbol{A},\boldsymbol{S}) = \frac{1}{2}\sum_{i=1}^{N}\|\boldsymbol{y}_i - \boldsymbol{A}\boldsymbol{s}_i\|_2^2 + \lambda_a\Omega(\boldsymbol{A}) + \lambda_s\Psi(\boldsymbol{S}), \qquad (2)$$

where $\Psi(\cdot)$ and $\Omega(\cdot)$ denote regularization functions, and $\lambda_a \geq 0$ and $\lambda_s \geq 0$ are regularizer coefficients. We use the squared Frobenius norm for regularization, which is the matrix variant of the ℓ_2 regularizer, *i.e.* $\Omega(\boldsymbol{A}) = \frac{1}{2}\|\boldsymbol{A}\|_{\mathrm{F}}^2 = \frac{1}{2}\sum_i\|\boldsymbol{a}_i\|_2^2 = \frac{1}{2}\sum_{ij}a_{ij}^2$, the sum of the squared matrix elements. Similarly for the VideoStory matrix $\Psi(\boldsymbol{S}) = \frac{1}{2}\|\boldsymbol{S}\|_{\mathrm{F}}^2$.

Predictability. The L_p function measures the occurred loss between the VideoStory $\boldsymbol{S}$ and the embedding of video features using $\boldsymbol{W}$. We define L_p as a regularized regression, similar to ridge regression:

$$L_p(\boldsymbol{S},\boldsymbol{W}) = \frac{1}{2}\sum_{i=1}^{N}\|\boldsymbol{s}_i - \boldsymbol{W}^{\top}\boldsymbol{x}_i\|_2^2 + \lambda_w\Theta(\boldsymbol{W}), \qquad (3)$$

where we use (again) the Frobenius norm for regularization of the visual projection matrix W, $\Theta(\boldsymbol{W}) = \frac{1}{2}\|\boldsymbol{W}\|_{\mathrm{F}}^2$, and λ_w is the regularization coefficient.

The VideoStory objective function, as given in Eq. (1), is convex with respect to matrix $\boldsymbol{A}$ and $\boldsymbol{W}$ when the embedding $\boldsymbol{S}$ is fixed. In that case, the joint optimization is decoupled into Eq. (2) and Eq. (3), which are both reduced to a standard ridge regression for a fixed $\boldsymbol{S}$. Moreover, when both $\boldsymbol{A}$ and $\boldsymbol{W}$ are fixed, the objective in Eq. (1) is convex w.r.t. $\boldsymbol{S}$. Therefore we use standard stochastic gradient descent by computing the gradients of a sample w.r.t. the current value of the parameters, and we minimize $\boldsymbol{S}$ jointly with $\boldsymbol{A}$ and $\boldsymbol{W}$.

To predict our VideoStory representation from a low-level video feature $\boldsymbol{x}_i$ we use

$$\boldsymbol{s}_i = \boldsymbol{W}^\top \boldsymbol{x}_i, \tag{4}$$

Then, using the predicted representation $\boldsymbol{s}_i$, the term vectors for each unseen video are predicted as:

$$\hat{\boldsymbol{y}}_i = \boldsymbol{A}\boldsymbol{s}_i = \boldsymbol{A}\boldsymbol{W}^\top \boldsymbol{x}_i, \tag{5}$$

where the words with the highest values are most relevant for this video.

To enable zero-example recognition, we employ the following steps: First, each test video is represented by predicting its term vector $\hat{\boldsymbol{y}}_i$ using Eq. (5), based on the pretrained embeddings. Second, we translate the textual event definition into the event query, denoted as $\boldsymbol{y}^e \in \mathbb{R}^M$, by matching the word2vec [15] mapping of the words in the event definition with the M unique words in the VideoStory dictionary. Finally, the zero-example ranking is obtained by measuring the similarity between the video representations and the event query based on the cosine similarity:

$$s_e(\boldsymbol{x}_i) = \frac{\boldsymbol{y}^{e\top} \hat{\boldsymbol{y}}_i}{||\boldsymbol{y}^e|| \quad ||\hat{\boldsymbol{y}}_i||}. \tag{6}$$

1.2 Representation II: ImageNet Shuffle

The second representation builds on concepts obtained after an ImageNet Shuffle [14]. We start from a Google inception network [17] trained on 22K ImageNet concepts. To deal with the problems of over-specific classes and classes with few images, we introduce a bottom-up and top-down approach for reorganization of the ImageNet hierarchy based on all its 21,814 classes and more than 14 million images. The classes in the ImageNet dataset are a subset of the WordNet collection and the classes are therefore connected in a hierarchy. The connectivity between classes provides information about their semantic relationship. We utilize the hierarchical relationship of WordNet for combining classes to generate reorganized ImageNet hierarchies for pre-training, as detailed in [14]. After this ImageNet Shuffle we maintain about 13k concepts. For event detection, we average the representations of the frames over each video, followed by ℓ_1-normalization.

1.3 Submissions

0Ex_baseline This 0ex baseline run uses the output of the probability layer of a Google inception net applied to two frames per second. A uniform filter is applied to the frame level output. The filter output is ranked per video based on cosine similarity to a vector model which consists of the top three concepts closest to the query terms in word2vec space. The cosine similarities are put through a percentile filter to determine the video score. Two CNN's are used. The first is trained on FCVID. The second is trained on a combination of the Fudan Columbia Video dataset (FCVID) [10], UCF-101 [16], and TRECVID. The final output is a late fusion of these two.

0Ex_topic In this 0ex run the text queries are represented using a topic model. Then, we learn to embed the CNN features of the videos into the topic model space. The final score is the cosine similarity between the embedded video and the represented text query. This is an early version of the approach presented in [8].

0Ex_combi This run combines the 0ex baseline with VideoStory. Where VideoStory uses as input feature the averaged output of the pool5 layer of a Google inception net applied to two frames per second. VideoStory translates this visual representation into words from its vocabulary. Videos are ranked based on cosine similarity to the word2vec mapping of the event text onto the VideoStory vocabulary. Three versions of VideoStory are used. The first is trained on the original YouTube46k dataset [6]. The second on FCVID. The third is trained on the combination of both datasets. The final VideoStory output is a late fusion of these three. The final output of the run is the fusion of the baseline plus the VideoStory fusion.

10Ex_baseline This 10ex run uses three modalities:

- *Low-level visual features:* The system computes the pool5 layer of a Google inception net on two frames per second. The features are averaged per video to obtain a video-level representation. A HIK SVM model is trained based on these features and used to classify videos.
- *High-level visual features:* The system applies a Google inception net trained on 12988 classes after an ImageNet Shuffle on two frames per second. The probabilities are averaged per video to obtain a video-level representation. A HIK SVM model is trained based on this representation and used to classify videos.
- *VideoStory*: The VideoStory transformation is applied to the low level visual features above. A HIK SVM model is trained based on the VideoStory embedding and used to classify videos.

The final output of the system is based on fusion of all three modalities.

10Ex_combi This 10ex run uses five modalities, the same ones as the 10ex_baseline, plus two additional ones:

- *Low level audio features:* The system computes a Fisher vector of MFCC coefficients and their first and second order derivatives. A HIK SVM model is trained based on these features and used to classify videos.
- *Low level motion features:* The system computes MBH and HOG descriptors along the motion trajectories.

Table 1: MediaMill multimedia event detection InfAP200% results on MED16EvalFull for 0ex and 10ex. Results in bold indicate the best runs among all submissions in TRECVID 2016 for this task with fully automatic condition.

	0ex	**10ex**	
Run		pre-specified	ad hoc
0Ex_baseline	13.5	–	–
0Ex_topic	11.1	–	–
0Ex_combi	**14.9**	–	–
10Ex_baseline	–	36.8	44.5
10Ex_combi	–	**39.4**	**46.3**

These local descriptors are then aggregated in each video using a Fisher vector to represent it. After normalization, a linear SVM is trained for each event and applied on the videos to obtain confidence scores.

The final output of the system is based on fusion of all five modalities.

1.4 Results

We summarize our results in Table 1.

2 Task II: Video Description

We also participated in the TRECVID 2016 showcase task of Video to Text Description, which consists of two subtasks, *i.e.*, Matching and Ranking, and Description Generation.

2.1 Matching and Ranking

In this subtask, participants were asked to rank a list of pre-defined sentences in terms of relevance for a given video. The test set consists of 1,915 videos collected from Twitter Vine. Each video is about 6 sec long. The videos were given to 8 annotators to generate a total of 3,830 sentences, with each video associated with two sentences written by two different annotators. The sentences have been split into two equal-sized subsets, set A and set B, with the rule that sentences describing the same video are not in the same subset. Per test video, participants are asked to rank all sentences in the two subsets.

Approach. We rely on Word2VisualVec as detailed in [4]. Briefly, our goal is to learn a visual representation from a natural language description. By doing so, the relevance between a given video x and a specific sentence q can be directly computed in a visual feature space. More formally, let $\phi(x) \in \mathbb{R}^d$ be a d-dimensional visual feature vector. We instantiate $\phi(x)$ with a ConvNet feature vector. We aim for a sentence representation $r(q) \in \mathbb{R}^d$ such that the similarity can be expressed in terms of $\phi(x)$ and $r(q)$, say, in the form of an inner product. Word2VisualVec is designed to produce $r(q)$. To handle sentences of varied length, we choose to first vectorize each sentence with a 500-dimensional word2vec model [15]. Following [2, 9, 11], we train word2vec on a corpus of Flickr tags rather than the web documents. Let $w2v(w)$ be individual word embedding vectors, we obtain the embedding vector of the input text by mean pooling over its words. The output of the first layer then goes through a multi-layer perceptron to produce $r(q)$, which resides in the visual feature space [4].

At training time, given an video x and a sentence q describing the video, we propose to reconstruct its visual feature $\phi(x)$ directly from q, with Mean Squared Error as our objective function and we solve it by stochastic gradient descent with RMSprop [18]. NIST provides a training set of 200 videos, which we consider insufficient for training Word2VideoVec. Instead, we learn the network parameters using video-text pairs from MSR-VTT [20], with hyper-parameters tuned on the provided TRECVID training set.

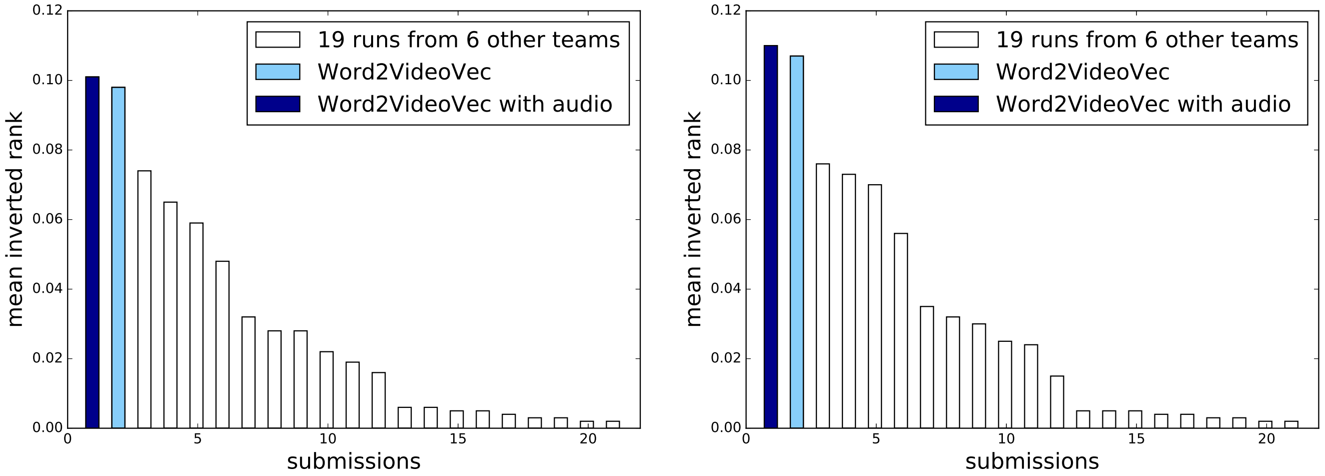
Submissions. We used the pre-trained GoogLeNet-shuffle model [14] to extract 1,024-dim visual features per individual frame. We obtain the video-level feature by meaning pooling, as suited for short videos. The audio channel of a video can sometimes provide complementary information to the visual channel. For instance, to help decide whether a person is talking or singing. To exploit this channel, we extract a 1,024-dim bag of quantized Mel-frequency Cepstral Coefficients vector [5] and concatenate it with the previous visual feature. Word2VisualVec is trained to predict such a visual-audio feature, as a whole, from input text. Word2VisualVec is used in a principled manner, transforming an input sentence to a video feature vector, let it be visual or visual-audio. For the sake of clarity we term the video variant *Word2VideoVec*.

Results. The performance metric is Mean Inverted Rank at which the annotated item is found. Higher mean inverted rank means better performance. As shown in Fig. 1, with Mean Inverted Rank ranging from 0.097 to 0.110, Word2VideoVec leads the evaluation on both set A and set B in the context of all submissions from seven teams worldwide. Moreover, the results can be further improved by predicting the visual-audio feature. We refer the interested reader to [4], where we present more experiments that detail the model's properties and capabilities for video (and image) to sentence matching.

2.2 Description Generation

In this subtask, participants were asked to generate a sentence to describe a specific test video, independently and without taking into consideration the existence of the sentence sets A and B.

Approach. We employ the *Early Embedding + Late Reranking* approach [3] for video description generation. This approach is built on top of a neural image captioning model [19], enhancing it with two novel modules. One is early embedding, which enriches the current low-level input to LSTM by tag embeddings. The other is late rerank-



(a) Results on the given set A

(b) Results on the given set B

Figure 1: State-of-the-art video-to-sentence matching results in the TRECVID 2016 benchmark, showing the good performance of Word2VideoVec compared to 19 alternative approaches, which can be further improved by predicting the visual-audio feature.

ing, for re-scoring generated sentences in terms of their relevance to a specific video. We try two distinct strategies to implement the tag embedding module. One is to employ a number of existing video taggers to automatically predict at most three tags for each video. See [3] for details. Word2VisualVec from the first task is re-employed to encode the predicted tags into a video feature vector. In the second strategy, based on the observation that pairs of tags are deemed to be more descriptive than individual tags [12], we extract common bi-grams from the MSR-VTT sentences [20], covering varied combinations of nouns, verbs and adjectives such as man_talk, young_girl, girl_singing, and playing_guitar. This results in a vocabulary of 288 bi-grams. An MLP classifier that predicts the bi-grams is trained on the MSR-VTT dataset. For each video the MLP output is used as another semantic-enriched representation, which we term Video Bi-gram Vector. The Word2VisualVec vector and the video bi-gram vector are used separately to initialize LSTM.

In addition, as some events usually happen in relatively fixed scenes, we heuristically append *where* at the end of the generated sentence if the events are detected. In particular, if the predicted sentence contains a specific sport word (basketball / baseball / football, etc), the phrase *on a sport field* is added. If the sentence contains sing or dance, the phrase *on a stage* is added. These naive rules result in a small performance gain.

Submissions. We use the same video feature, *i.e.*, GoogleNet-shuffle, as used in the matching and ranking subtask. The network structure of the MLP is 1024×1024×288.

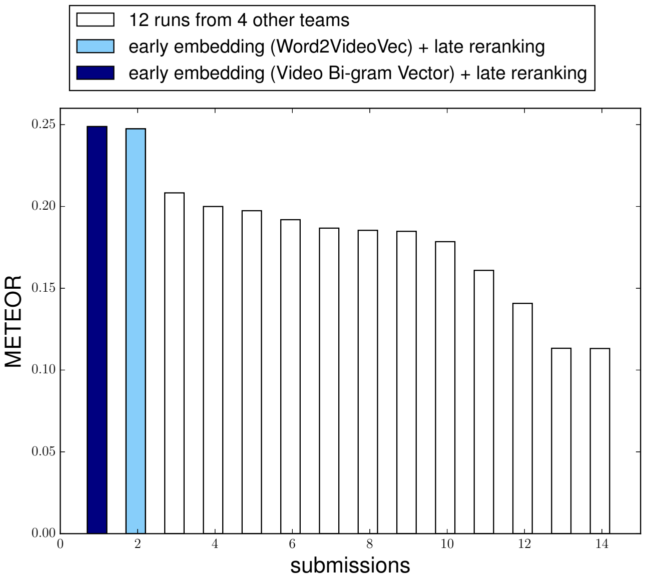


Figure 2: State-of-the-art video description generation result in the TRECVID 2016 benchmark, showing the good performance of our Early Embedding + Late Reranking solution compared to 12 approaches.

Results. The performance metric is METEOR, higher is better. As shown in Fig. 2, our runs perform the best compared to submissions from the other participants in this subtask. Specifically, early embedding with Video Bi-gram Vector (METEOR of 0.2488), is marginally better than early embedding with Word2VisualVec(METEOR of 0.2475). The result is encouraging as tag embedding can be simplified without hurting performance.

Acknowledgments

The authors are grateful to NIST and the TRECVID coordinators for the benchmark organization effort. The University of Amsterdam acknowledges support by the STW STORY project. The Renmin University of China acknowledges support by NSFC (No. 61672523).

References

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

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

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

[4] J. Dong, X. Li, and C. G. M. Snoek. Word2VisualVec: Image and video to sentence matching by visual feature prediction. *CoRR*, abs/1604.06838, 2016.

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

[6] A. Habibian, T. Mensink, and C. G. M. Snoek. Videostory: A new multimedia embedding for few-example recognition and translation of events. In *MM*, 2014.

[7] A. Habibian, T. Mensink, and C. G. M. Snoek. Video2vec embeddings recognize events when examples are scarce. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. In press.

[8] N. Hussein, E. Gavves, and A. W. M. Smeulders. Unified embedding with metric learning for zero-exemplar event detection. In *CVPR*, 2017.

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

[10] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, and S.-F. Chang. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *CoRR*, abs/1502.07209, 2015.

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

[12] X. Li, C. G. M. Snoek, M. Worring, and A. W. M. Smeulders. Harvesting social images for bi-concept search. *IEEE Transactions on Multimedia*, 14(4):1091–1104, Aug. 2012.

[13] M. Mazloom, X. Li, and C. G. M. Snoek. TagBook: A semantic video representation without supervision for event detection. *IEEE Transactions on Multimedia*, 18(7):1378–1388, 2016.

[14] P. Mettes, D. Koelma, and C. G. M. Snoek. The imagenet shuffle: Reorganized pre-training for video event detection. In *ICMR*, 2016.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[16] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.

[17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

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

[19] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015.

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