VideoStory: A New Multimedia Embedding for Few-Example Recognition and Translation of Events

Amirhossein Habibian, Thomas Mensink, Cees G. M. Snoek
ISLA, Informatics Institute, University of Amsterdam
Science Park 904, 1098 XH
Amsterdam, The Netherlands
{a.habibian, tmensink, cgmsnoek}@uva.nl

ABSTRACT

This paper proposes a new video representation for few-example event recognition and translation. Different from existing representations, which rely on either low-level features, or pre-specified attributes, we propose to learn an embedding from videos *and* their descriptions. In our embedding, which we call *VideoStory*, correlated term labels are combined if their combination improves the video classifier prediction. Our proposed algorithm prevents the combination of correlated terms which are visually dissimilar by optimizing a joint-objective balancing descriptiveness and predictability. The algorithm learns from textual descriptions of video content, which we obtain for free from the web by a simple spidering procedure. We use our VideoStory representation for few-example recognition of events on more than 65K challenging web videos from the NIST TRECVID event detection task and the Columbia Consumer Video collection. Our experiments establish that *i)* VideoStory outperforms an embedding without joint-objective and alternatives without any embedding, *ii)* The varying quality of input video descriptions from the web is compensated by harvesting more data, *iii)* VideoStory sets a new state-of-the-art for few-example event recognition, outperforming very recent attribute and low-level motion encodings. What is more, VideoStory translates a previously unseen video to its most likely description from visual content only.

Categories and Subject Descriptors

I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—*Video analysis*

General Terms

Algorithms, Experimentation, Measurement

1. INTRODUCTION

The goal of this paper is to recognize *and* translate events in web video from ten examples only. For multimedia challenges where examples are scarce, a traditional solution is to rely on clever combinations and rerankings of as many extracted information sources as possible *e.g.,* [25,28]. In principle, such approaches would be well-suited for event recognition, despite the fact that none of them is able to provide a semantic interpretation of the obtained result. We adhere to a different solution. The main contribution of this paper is a new video representation for few-example event recognition and translation, which we learn from freely available web videos and their descriptions. We call our new video representation *VideoStory*, since it strives to encode the story of a video rather than a collection of single words (Figure 1). Before detailing our contributions we discuss related work on representations for event recognition, learning from the web, and multimedia embedding.

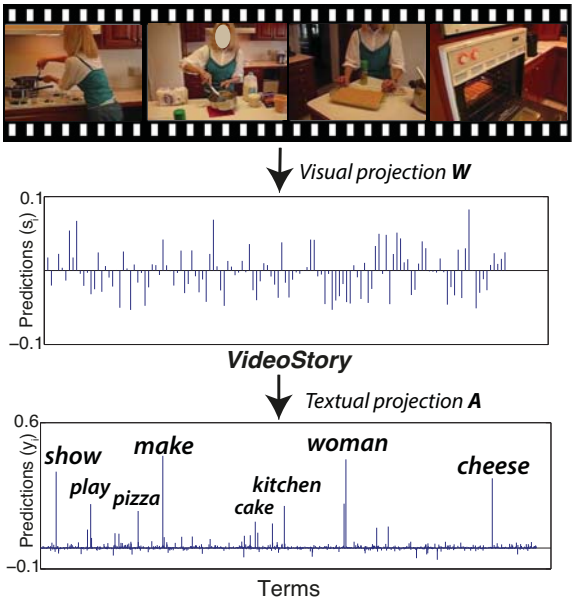

Figure 1: VideoStory at work. We propose an algorithm that learns from web data an optimal multimedia embedding optimizing both the visual projection for recognition and the textual projection for translation. For unseen videos it recognizes events from just ten examples and it predicts the most likely description from visual content only.

Representations for Event Recognition The state-of-the-art in event recognition represents a video in terms of

MM'14, November 3–7, 2014, Orlando, Florida, USA.

ACM 978-1-4503-3063-3/14/11 ...$15.00.
http://dx.doi.org/10.1145/2647868.2654913.

low-level audiovisual features [2,6,16,27,34,37]. These methods first extract from the video various descriptors *e.g.,* color SIFT [36], or motion boundary histograms [37]. Then, the descriptors are quantized and aggregated as bag-of-word histograms or Fisher vectors [32,37]. Despite good recognition performance, low-level representations suffer from two drawbacks. First, they are incapable of providing a semantic interpretation of an event. Second, because of their high-dimensionality, training effective event classifiers on the low-level representation often requires a sufficient number of training examples. When only a few event exemplars are available, the applicability of the low-level representation is limited [22, 23].

Semantic video representations provide an alternative, they are achieved by representing a video by their attribute (or concept) scores. Yang *et al.* [41] obtain the attributes by compressing their low-level features in three consecutive unsupervised clustering steps. The obtained attributes are shown to outperform the low-level audiovisual features they started from. However, attributes obtained by unsupervised clustering of low-level features still have no semantic interpretation. In [22], Ma *et al.* learn the attribute classifiers and event classifiers jointly. In their work, the attribute classifiers are trained to optimize the event recognition, without explicitly optimizing the individual attribute classifier accuracies. As a consequence, the obtained representation does not necessarily have a semantic interpretation either. We aim for a representation that is both robust for recognition and interpretable by humans.

Semantic representations for event recognition that are both accurate and interpretable rely on the prediction scores made by a set of pre-trained attribute classifiers [11, 13, 21, 22, 24, 26]. In [26], Merler *et al.* obtain the attribute classifiers by training 280 SVMs on a manually labeled collection of images, including objects, people, and scenes. Habibian *et al.* [11] study the properties of 1,346 attribute classifiers trained from ImageNet [8] and TRECVID [29] for representing and recognizing events in web video. In [24], Mazloom *et al.* learn a semantic representation per event by feature selection of a subset of attributes, from the same 1,346 used in [11], which maximizes the event recognition accuracy. While effective and promising, a considerable drawback of attribute representations is their dependence on individual attribute classifiers. They demand a considerable classifier training and video representation construction effort. However, the biggest limitation is the need to prespecify and manually label the attributes in advance, often leading to a mismatch between the representation and the events of interest. Rather than specifying and labeling the attributes one by one to represent video, we propose an algorithm that *discovers* the semantic representation without the need for specification or annotation.

Representation Learning from the Web Many have used annotations from the web to learn individual attributes, *e.g.,* [17, 35, 39], without considering their utility as a representation. We are inspired by the work on image representations by Berg *et al.* [4], who start by collecting a set of web pages, relevant to the (visual) attributes of interest, by submitting attribute names to an Internet search engine. The retrieved web pages are mined to discover the most frequent terms and their associated images. Subsequently, a classifier is trained by using all the images corresponding to a certain term as positive annotations and randomly sampling images from other terms as negatives. They hold out a portion of the training data as a validation set, which they use to select the most reliable term classifiers for the final representation. While their work is proposed for images, it is easily extended to video. However, a drawback is that many terms rarely occur. For these infrequent terms only a limited number of positive examples are available, which leads to a biased estimation of their reliability. As a consequence, many of the discovered visual terms might be overfitted to their small training set and do not generalize well for new videos. We also discover our representation from the Internet, but rather than selecting individual, and often unreliable, classifiers per term, we prefer to combine terms automatically into more descriptive attributes. By combining terms, more training examples are available and a more robust representation is obtained, without losing descriptive ability.

Representation by Multimedia Embedding Current representations for video (and image) translation also combine terms, typically by multimedia embeddings of visual features and descriptions from labels [40], attributes [1] or documents [7,31] into a joint low dimensional space. Most multimedia embeddings are proposed for image classification [1, 40] and cross-media retrieval [31], we are not aware of any multimedia embeddings for few-example event recognition and translation. Das *et al.* [7] focus on generating video translations. They model the relation between low-level video-features and textual terms by an embedding founded on multimodal latent dirichlet allocation; a probabilistic topic model which models the videos and their descriptions as mixed memberships over a set of latent topics. Despite their effectiveness for translating video to text, this method is not intended to recognize events in video. As an aside we note that by design multimodal topic models are only applicable for discrete features. Therefore, they cannot leverage recent state-of-the-art video encodings such as Fisher vectors [32] or deep learned representations [19], which would undermine the effectiveness of multimodal topic models for the purpose of event recognition. In addition to generating event video translations, we aim for state-of-the-art video event recognition using a multimedia embedding that leverages the benefits of modern encodings.

Contributions We make the following contributions:

- We propose a multimedia embedding for few-example event recognition and translation, which we learn from videos and their descriptions.
- We introduce an algorithm that combines correlated terms if their combination improves the video classifier prediction by optimizing a joint-objective trading of descriptiveness and predictability (Section 2).
- The algorithm learns from textual descriptions of video content, for which we introduce a new dataset obtained for free from the web by a simple spidering procedure (Section 3).

We use our VideoStory representation for few-example recognition and translation of events on more than 65K challenging web videos from the NIST TRECVID event detection task [33] and the Columbia Consumer Video collection [15] (Section 4). State-of-the-art results support our proposed contributions (Section 5). We conclude in Section 6.

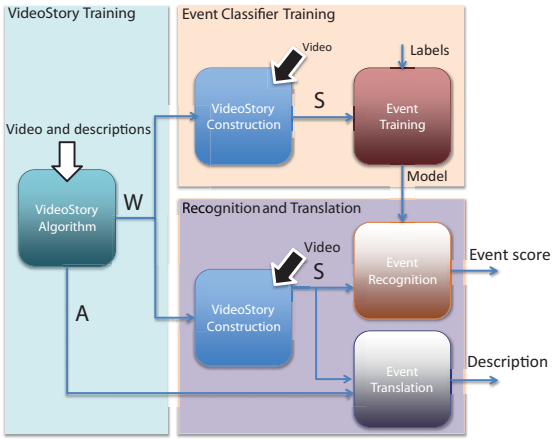

Figure 2: Dataflow for learning the VideoStory and using it for event recognition and translation.

2. VIDEOSTORY FRAMEWORK

Our VideoStory framework contains three major parts, schematically illustrated in Figure 2.

1. The VideoStory training, where we learn our multimedia embedding from a dataset consisting of videos with descriptions. This training outputs two projection matrices: a visual projection matrix $\boldsymbol{W}$, and a textual projection matrix $\boldsymbol{A}$. The VideoStory representation $\boldsymbol{S}$ is computed from the visual projection matrix $\boldsymbol{W}$ and low-level video features.

2. The event classifier training, where we use off-the-shelf SVMs to train classifiers on a dataset consisting of videos with a few event labels. The videos are encoded with our VideoStory representation.

3. The recognition and translation stage, where we evaluate the event classifiers, and use the semantics of our representation to describe videos.

In this section we introduce the VideoStory embedding, its design principles and how it is obtained by learning.

2.1 Objective Function

Using the notation summarized in Table 1, we will describe the objective function we minimize to obtain the VideoStory representation. To learn the embedding we use a dataset of videos, represented by low-level video features $\boldsymbol{X}$, and their descriptions, represented by binary term vectors $\boldsymbol{Y}$, indicating which terms are present in each video description. While we use and emphasize low-level visual features in this work, our approach is generic and can create a VideoStory from any multimedia feature.

The aim of the VideoStory representation is to balance two compelling forces:

1. *Descriptiveness*, to preserve the information encoded in the video descriptions $\boldsymbol{Y}$ as much as possible, and

2. *Predictability*, to ensure that the VideoStory could be effectively recognized from visual video content $\boldsymbol{X}$.

Notation	Description
N	Number of videos
M	Number of unique terms in descriptions
D	Dimensionality of visual feature
k	Dimensionality of VideoStory embedding
$\boldsymbol{X} \in \mathbb{R}^{D \times N}$	Matrix of low-level video features
$\boldsymbol{Y} \in \{0,1\}^{M \times N}$	Matrix of binary term vectors
$\boldsymbol{W} \in \mathbb{R}^{D \times k}$	VideoStory visual projection
$\boldsymbol{A} \in \mathbb{R}^{M \times k}$	VideoStory textual projection
$\boldsymbol{S} \in \mathbb{R}^{k \times N}$	VideoStory embedding
$\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{s}_i$	The column representing the i-th video

Table 1: Summary of notation.

Therefore, we learn the VideoStory representation by both objectives in a joint optimization framework.

The VideoStory representation is learned by minimizing:

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

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 VideoStory embedding $\boldsymbol{S}$ interconnects the two loss functions. To the best of our knowledge this joint embedding framework is novel.

Descriptiveness For the L_d function, we use a variant of regularized Latent Semantic Indexing [38]. This objective minimizes the quadratic error between the original video descriptions $\boldsymbol{Y}$, and the reconstructed translations obtained from $\boldsymbol{A}$ and $\boldsymbol{S}$:

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

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.,* $\Omega(\boldsymbol{A}) = \|\boldsymbol{A}\|_{\text{F}}^2 = \sum_i \|\boldsymbol{a}_i\|_2^2 = \sum_{ij} a_{ij}^2$, the sum of the squared matrix elements. Similarly for the VideoStory matrix $\Psi(\boldsymbol{S}) = \|\boldsymbol{S}\|_{\text{F}}^2$.

The main difference with regularized Latent Semantic Indexing [38] is that they used an ℓ_1 regularizer, $\Omega(\boldsymbol{A}) = \sum_i \|\boldsymbol{a}_i\|_1$, which enforces sparsity in the textual projection $\boldsymbol{A}$. However, with our larger representation (typically we use k between 256 and 1,024 in our experiments compared to only $k = 20$ used in [38]) and fewer number of unique terms (around 10K, compared to 100K), enforcing sparsity is not necessary for good performance.

Note that many other textual embedding methods, such as Sparse Coding and probabilistic Latent Semantic Indexing [12] can be formulated similar to Eq. (2), when appropriate regularization functions $\Omega(\cdot)$ and $\Psi(\cdot)$ are used. Furthermore, when the textual projection matrix $\boldsymbol{A}$ is constrained such that each column has a single non-zero value, *i.e.,* selects a single term, our objective becomes very close to methods that select the best single term labels, such as [4].

Predictability The L_p function measures the occurred loss between the VideoStory $\boldsymbol{S}$ and the embedding of low-level videos features using $\boldsymbol{W}$. Since the VideoStory $\boldsymbol{S}$ is real valued, as opposed to a binary or multi-class encod-

ing, we can not rely on standard classification losses such as the hinge-loss used in SVMs. Therefore, we define L_p as a regularized regression, similar to ridge regression:

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

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

2.2 Learning Algorithm

To handle large scale datasets and state-of-the-art high-dimensional visual features, *e.g.,* Fisher vectors [32] on low-level video features [37] or deep learned representations [19], we employ SGD (Stochastic Gradient Descent) [5]. SGD is an efficient online procedure and converges fast to the (global) minimum of a model. At each step, training with SGD consists of (i) choosing a random sample from the dataset consisting of a video and a description, (ii) computing the sample estimate of the gradient of the parameters in the model, and (iii) updating the parameters in the direction of the gradient with step-size η. The number of passes over the datasets, often denoted as *epochs*, and the step-size η are hyper-parameters of SGD.

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 Eq. (1) is convex *w.r.t.* $\boldsymbol{S}$. Therefore we use standard SGD 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 $\boldsymbol{W}$.

Lets denote a randomly sampled video and description pair at step t by $(\boldsymbol{x}_t, \boldsymbol{y}_t)$, and let $\boldsymbol{s}_t$ denote the current VideoStory embedding of sample t. The gradients of Eq. (1) for this sample *w.r.t.* $\boldsymbol{A}, \boldsymbol{W}$ and $\boldsymbol{s}_t$ are given by:

$$\nabla_{\boldsymbol{A}} L_{\mathrm{VS}} = -2\left(\boldsymbol{y}_t - \boldsymbol{A}\boldsymbol{s}_t\right)\boldsymbol{s}_t^\top + \lambda_a \boldsymbol{A}, \quad (4)$$

$$\nabla_{\boldsymbol{W}} L_{\mathrm{VS}} = -2\ \boldsymbol{x}_t \left(\boldsymbol{s}_t - \boldsymbol{W}^\top \boldsymbol{x}_t\right)^\top + \lambda_w \boldsymbol{W}, \text{ and} \quad (5)$$

$$\nabla_{\boldsymbol{s}_t} L_{\mathrm{VS}} = \ \ 2\left[\boldsymbol{s}_t - \boldsymbol{W}^\top \boldsymbol{x}_t - \boldsymbol{A}^\top\left(\boldsymbol{y}_t - \boldsymbol{A}\boldsymbol{s}_t\right)\right] + \lambda_s \boldsymbol{s}_t. \quad (6)$$

Our algorithm is summarized in Algorithm 1.

The effect of joint learning the descriptiveness and the predictability, becomes clear in Eq. (6), where both the textual projection matrix $\boldsymbol{A}$ and visual projection matrix $\boldsymbol{W}$ contribute to learning the VideoStory embedding $\boldsymbol{S}$. This embedding $\boldsymbol{S}$ is subsequently used to obtain the textual projection $\boldsymbol{A}$ matrix, in Eq. (4), and the visual projection $\boldsymbol{W}$ matrix, in Eq. (5). This leads to the VideoStory embedding, which is both descriptive, by preserving the textual information, and predictable, by minimizing the visual prediction loss.

2.3 Using the VideoStory Embedding

The result of training our VideoStory embedding is the visual projection matrix $\boldsymbol{W}$ and the textual projection matrix $\boldsymbol{A}$. These are used to encode a new video i into our VideoStory representation $\boldsymbol{s}_i$.

In the case that both a video $\boldsymbol{x}_i$ and description y_i are given, we could obtain the semantic embedding by returning $\boldsymbol{s}_i$ from Eq. (1), while keeping both $\boldsymbol{A}$ and $\boldsymbol{W}$ fixed.

input : $\boldsymbol{X}$, $\boldsymbol{Y}$, k, η (step-size), m (max-epochs)
output: $\boldsymbol{W}$ and $\boldsymbol{A}$

$\boldsymbol{A}$, $\boldsymbol{W}$, and $\boldsymbol{S} \leftarrow$ random (zero-mean, unit variance)
for $e \leftarrow 1$ ***to*** m **do**
  for $i \leftarrow 1$ ***to*** N **do**
    Pick a random video-description pair $(\boldsymbol{x}_t, \boldsymbol{y}_t)$
    Compute gradients *w.r.t.* $\boldsymbol{A}, \boldsymbol{W}$ and $\boldsymbol{s}_t$
    Update parameters:
      $\boldsymbol{A} \leftarrow \boldsymbol{A} - \eta_t \nabla_{\boldsymbol{A}} L_{\mathrm{VS}}$ see Eq. (4)
      $\boldsymbol{W} \leftarrow \boldsymbol{W} - \eta_t \nabla_{\boldsymbol{W}} L_{\mathrm{VS}}$ see Eq. (5)
      $\boldsymbol{S} \leftarrow \boldsymbol{s}_t - \eta_t \nabla_{\boldsymbol{s}_t} L_{\mathrm{VS}}$ see Eq. (6)
  end
end
return: $\boldsymbol{W}$ and $\boldsymbol{A}$

Algorithm 1: Pseudocode for learning VideoStory

However, in practice most videos are not provided with a description. Therefore, we use:

$$\boldsymbol{s}_i = \boldsymbol{W}^\top \boldsymbol{x}_i, \quad (7)$$

to construct our VideoStory representation from the low-level video features $\boldsymbol{x}_i$. Given an embedded video $\boldsymbol{s}_i$, we can translate a video by:

$$\hat{\boldsymbol{y}}_i = \boldsymbol{A}\boldsymbol{s}_i, \quad (8)$$

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

3. HARVESTING VIDEOS AND THEIR DESCRIPTIONS FROM THE WEB

Rather than describing the video content manually, we opt to harvest both videos and descriptions from the web. Video sharing web sites, such as YouTube and Vimeo, provide a rich and varied source of videos and user provided descriptions, such as their title captions and comments. Although video title captions do not necessarily correspond to the visual content of the videos, we will show that by harvesting a large number of these captioned videos and applying a set of quality filters we obtain reliable video descriptions.

We start from an initial pool of descriptions, as the collection seeds, and iteratively collect videos and their title captions from YouTube. For the collection seeds, we rely on 3,000 sentence descriptions from the training partition of the NIST TRECVID HAVIC corpus [33]. Then each description within the pool is queried to YouTube and the 25 most relevant videos are retrieved, based on YouTube's textual similarity search. Every retrieved video is passed through a set of quality filters. The videos which pass all the filters are added to the collection and their title captions are added to the description pool. We iteratively repeat this procedure until enough videos are collected. We will first detail our quality filters before providing the statistics of our harvested video and description dataset.

3.1 Quality Filters

Event Filter Events are generally described by their actors, actions, and possible involved objects [10]. Hence we assume that a description of an event video should contain actors, actions and objects. For this purpose, we parse the

Figure 3: Terms from the VideoStory46K dataset occurring in more than 500 title captions of the harvested YouTube videos.

grammatical structure of title captions using a probabilistic context free grammar parser [18]. Then we accept a video only if its caption includes verbs, subjects, and objects.

Visualness Filter There are many terms in title captions which do not refer to visually depictable attributes, such as `buy`, `God`, and `genius`. These attributes are not recognizable by present-day visual classifiers, so should not be included in the collection. For this filter we evaluate the visualness of caption terms. Rather than relying on visual features [4, 9, 20], which are expensive to extract and limit the scalability, we evaluate the visualness of each term in the title caption by measuring its similarities to the ImageNet synsets [8] in the WordNet hierarchy. We measure the similarity of each synset pair by following [3], which finds overlaps between the glosses of two synsets as well as their directly linked synsets. Finally, we define the visualness of a title caption by averaging the visualness of all its terms. The captions whose visualness exceeds a threshold of 0.5 are accepted by our harvesting procedure.

Reality Filter A considerable amount of YouTube videos are related to celebrities, TV series, and movie trailers. We observe these professional videos are typically semantically dissimilar to the event videos which we are interested in. Moreover, they often infringe intellectual property rights. Therefore, we prefer to filter out the corresponding videos. Our reality filter relies on a list of keywords from Wikipedia, which provides an extensive index of celebrity, TV series and movie names[1]. We exclude the videos whose description matches any of the keywords from the list.

Temporal Filter Our last filter rests on the assumption that short videos better match their title captions, compared to long videos. It is because long videos usually contain a broad set of attributes that are typically not specified completely by their captions. Hence, we only retrieve the YouTube videos which are shorter than 120 seconds.

3.2 VideoStory46K Dataset

Following the proposed procedure, including all quality filters, we harvest 45,826 videos from YouTube. The videos have an average length of 58.4 seconds and the whole collection contains 743 hours of videos. Every video comes with a short title caption provided by the user who has uploaded the video. Every caption is made of 7.7 individual terms on average, with a standard deviation of 1.8 terms. There are 19,159 unique terms in the captions, most of them occurring infrequently in the collection, *i.e.,* 50% of the terms occur only once in the collection, and only 0.4% of the terms occur more than 500 times. Some examples of these frequent terms are shown as a tag cloud in Figure 3. Our dataset of videos and their descriptions, which we call VideoStory46K, is available for download at `http://www.mediamill.nl`. Illustrative examples from the dataset are shown in Figure 4.

4. EXPERIMENTAL SETUP

4.1 Evaluation Datasets

We perform our experiments on the challenging NIST TRECVID HAVIC corpus [33] and the Columbia Consumer Video collection [15], together containing more than 65K videos collected from the web. To the best of our knowledge these are the largest publicly available video corpora in the literature for event recognition containing user-generated video with a large variation in quality, length and content.

NIST TRECVID HAVIC [33] The 2013 public release of this dataset comes with five partitions of videos: Event Kit training, Background training, test set MED, test set Kindred, and a Research collection, including about 200, 5K, 27K, 14K, and 10K videos, respectively[2]. Apart from the Research partition (which we only use in experiment 2), all four other partitions come with ground truth annotation at video-level for 20 event categories. We follow the *10Ex evaluation procedure* outlined by the NIST TRECVID event recognition task [29] for all our experiments. It means that for each event the training data is composed of 10 positive videos from the Event Kit training data along with about 5K negative videos from the Background training data. We report event recognition results of each event classifier on both the test set MED and test set Kindred datasets.

Columbia Consumer Video [15] This dataset contains 9,317 user-generated videos from YouTube. It consists of over 210 hours of videos in total, where each video has an average length of 80 seconds. Moreover, the dataset contains ground truth annotations at video-level for 20 semantic categories, where 15 of them are events. The other 5 categories are objects and scenes, which are excluded from the dataset in our experiments: "bird", "cat", "dog", "beach" and "playground". We use the standard partitioning of the dataset, but we use only 10 positive examples per event in the training data. These 10 are selected based on alphabetical order of the respective video names, we ignore the remaining positive examples in the train set. We report event recognition results on the standard test partition.

4.2 Event Recognition Protocol

Our event recognition pipeline consists of the following consecutive steps:

1. Extracting low-level features. For all videos we compute MBH descriptors [37] along the motion trajectories. The extracted 288-dimensional descriptors are reduced to 128 dimensions using PCA and are then aggregated per video using a Fisher vector [32], with 128 Gaussians resulting in a 32K dimensional vector. Each Fisher vector is power

[1] `wikipedia.org/wiki/List_of_American_television_series`

[2] There is also a PROGRESS set with 98K videos, but this partition is for blind testing by NIST only.

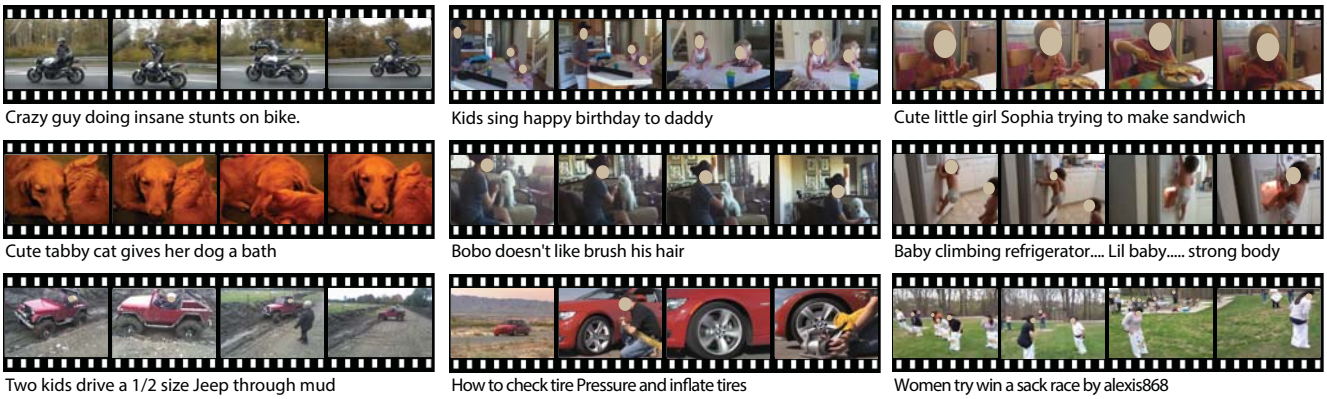

Figure 4: Example videos and title captions from our introduced VideoStory46K dataset.

normalized, with $\alpha = 0.2$, as in [14]. This representation is shown to be state-of-the-art for recognizing events using a single modality [37].

2. Learning the VideoStory. We learn the VideoStory on the MBH-encoded videos from the VideoStory46K dataset using the algorithm described in Algorithm 1, we use 75% of the dataset for training and 25% for validation to set the hyper-parameters of our model ($\lambda_w, \lambda_a, \lambda_s$) and of SGD (number of epochs, η).

3. Applying the VideoStory. We apply the learned VideoStory on all the MBH-encoded training and test videos in the NIST TRECVID HAVIC and Columbia Consumer Video datasets.

4. Training event classifiers. On top of the VideoStory we train event classifiers using the 10-example *training sets* of the NIST TRECVID HAVIC and Columbia Consumer Video datasets. We train SVM classifiers with RBF kernels, which is shown to be effective for learning events from semantic representations [26]. We obtain the parameters for the SVM regularization cost and the RBF kernel by 2-fold cross-validation.

5. Testing event classifiers. We apply the event classifiers on the VideoStory embedding of the MED test, Kindred test and Columbia Consumer Video test sets, and rank the video classification results.

As an evaluation criteria for the ranked lists, we follow the standard convention in the literature [15, 29] by relying on the average precision (AP) per event, and we report the mean average precision (mAP) for overall accuracy.

4.3 Experiments

4.3.1 Effect of Embedding

In our first experiment we quantify the merit of the VideoStory, by comparing it with three baselines: an embedding without joint optimization and two baselines without embedding inspired by the work on image representations by Berg *et al.* [4].

Baseline 1: ***Description embedding***. In this baseline we learn the embedding in two stages. First, the descriptions are embedded using regularized Latent Semantic Indexing [38], according to Eq. (2). Then the video embedding is learned separately, by minimizing the error of predicting the embedded descriptions from the videos using ridge regression, according to Eq. (3).

Baseline 2: ***Visual terms***. We learn this representation directly from the terms in the description following Berg *et al.* [4] (see Section 1). A linear SVM classifier is trained per term. The classifiers which have the highest prediction accuracy, based on a 2-fold cross-validation, are selected as visual terms in the video representation.

Baseline 3: ***Frequent terms***. This representation is similar to baseline 2, but rather than using cross-validation to select terms it simply selects the terms with the highest frequency in the descriptions. The occurrences of these terms in the description accompanying a video are considered as labels which are used for training the classifiers.

We evaluate all four video representations for recognizing events from few-examples using a varying dimensionality of the representation, from 32 up to 8,192.

4.3.2 Description Quality and Quantity

In our second experiment we assess the influence of the quality and quantity of the videos and descriptions that we use as input to learn our VideoStory. We compare the VideoStory learned from the VideoStory46K dataset with two baselines.

Baseline 4: ***ExpertSentences10K***, includes 10K videos from the Research partition of the NIST TRECVID HAVIC corpus [33]. Each video in this collection comes with an expert written description. The descriptions are made of a few sentences written by a team of 60 expert annotators with the purpose of summarizing the visual content of the videos. Consequently, there is always a strong correspondence between a description and its video in this collection.

Baseline 5: ***VideoStory10K***, includes 10K random videos and descriptions from the VideoStory46K dataset, which we collected as discussed in Section 3. This collection includes the same number of videos and captions as the ExpertSentences10K dataset, but the captions are generally of lower quality for event recognition because of the non-expert descriptions and the fact that video captions on YouTube do not necessarily correspond to the visual content.

Again we use a varying dimensionality of the representation, starting from 32 up to 2,048, and we compare their effectiveness for recognizing events from few examples.

4.3.3 VideoStory vs Others

In this experiment we compare the VideoStory with alternative representations for event recognition. We consider three state-of-the-art baselines.

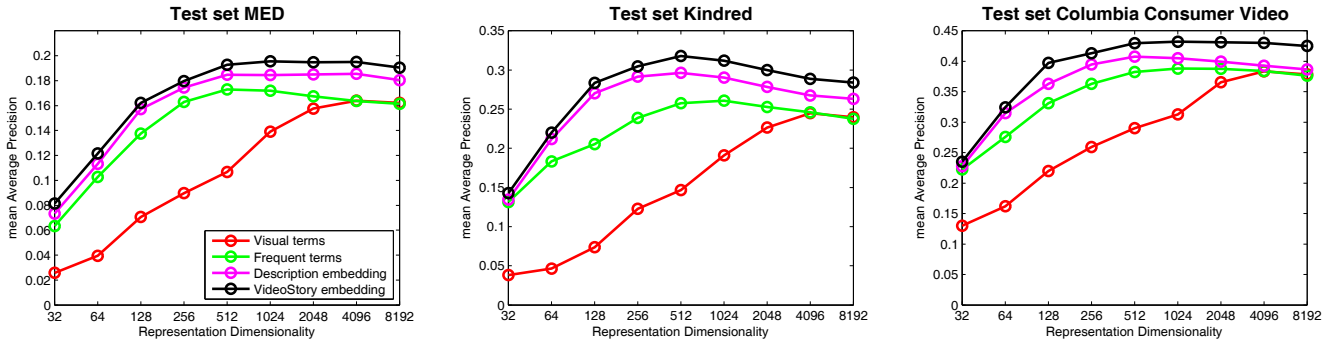

Figure 5: Effect of embedding. The VideoStory, which is learned by our proposed algorithm, outperforms all three alternatives on all three test sets. The Description embedding is the closest competitor, but it suffers from the embedding of correlated terms which are visually dissimilar. Representations without term label combinations, *i.e.,* the Visual terms and Frequent terms, inspired by Berg *et al.* [4], are always worse.

Baseline 6: ***Attributes*** [11]. This representation uses 1,346 pre-specified attribute classifiers to represent a video. Every image and key frame is represented as a Fisher vector encoding of densely sampled color SIFT descriptors [36] with spatial pyramids. Each individual attribute classifier is trained by a linear SVM on annotated images from TRECVID and ImageNet. The video representation is obtained by applying the trained classifiers on the video frames, extracted every two seconds, and then averaging over the entire video [11].

Baseline 7: ***Informative Attributes*** [24]. This baseline automatically selects informative attributes per event, and uses the selected subset as video representation. The informative attributes are selected from the same 1,346 attribute classifiers used for *baseline 6* by using mRMR feature selection [30] on the training data, with 2-fold cross validation. The reported results are based on the optimal number of selected attributes per event.

Baseline 8: ***Low-Level*** [37]. The last baseline is the state-of-the-art Fisher vector representation using MBH descriptors, explained in Section 4.2. In this case the event classifiers are trained directly on the low-level video representations, without extracting an embedding.

Different from the low-level baseline, the attributes and the informative attributes baselines rely only on static visual features, since they are trained, in part, on the ImageNet dataset. These baselines are included because of their translation ability, a capability the low-level features cannot achieve. For this experiment we use a fixed 1,024 dimensional VideoStory representation, since this resembles most closely with the dimensionality of the other semantic baselines (using at most 1,346 dimensions), although it is not necessarily the optimal VideoStory dimensionality.

5. RESULTS

5.1 Effect of Embedding

The results are shown in Figure 5. The VideoStory, which is learned by our proposed algorithm, outperforms all three alternatives on all three test sets.

The lowest performing representation is obtained with Visual terms, which relies on the estimated reliability of individual term classifiers. As expected, this representation suffers from two drawbacks. First, many of the visual terms refer to very specific terms, which are incapable of characterizing the events of interest *i.e.,* `necklace`, `suitcase`, `cellphone`, `elevator` and `earring`. Although these terms can be accurately predicted from videos, they are incapable of providing a characteristic representation of the events. Second, many of the terms rarely occur in video descriptions. For these infrequent terms there are only a limited number of positive examples available, which leads to a biased estimation of their reliability. As a consequence, many of the discovered visual terms might be overfitted to their small training set and do not generalize well for new videos.

The drawbacks of Visual terms are relaxed by Frequent terms, by simply relying on the most frequent terms. We observe the most frequent terms usually refer to characterizing attributes of events which are frequently used by humans when describing a video, *i.e.,* `car`, `girl`, `man`, `kid`, and `truck`. Moreover, because of their large number of positive examples, the trained visual classifiers are in general more reliable. Hence, Frequent terms consistently outperform Visual terms on the MED test, the Kindred test, and the Columbia Consumer Video datasets.

Both the Visual terms and Frequent terms use no embedding. However, a Description embedding, without joint optimization, is always better than using no embedding at all. We explain it by the fact that the Description embedding represents the terms in a reduced-dimensional space, where the correlated terms are usually combined together. Combining correlated terms per dimension leads to less correlation between dimensions. Moreover, as the positive examples for all correlated terms are combined, it provides more positive video examples to train visual classifiers, often leading to better accuracy. The results demonstrate that a more effective video representation is obtained by considering correlated terms together.

Finally, our proposed VideoStory outperforms the Description embedding, especially for higher dimensional representations. For example, by extracting a 1,024 dimensional Description embedding we obtain an event recognition mAP of 0.183, 0.287 and 0.405 on MED test, Kindred test, and Columbia Consumer Video test, respectively. However, by extracting a VideoStory of the same dimensionality we obtain an event recognition mAP of 0.196, 0.312, and 0.432, which is a relative improvement of 7%, 9%, and 7%. Given the difficulty of recognizing events from only ten positive examples [29], this is a considerable improvement. We explain the improvement by the fact that combining the terms based on textual correlation only does not necessarily imply that the corresponding video is visually correlated as well. For example, the terms `puppy` and `kid` have a high

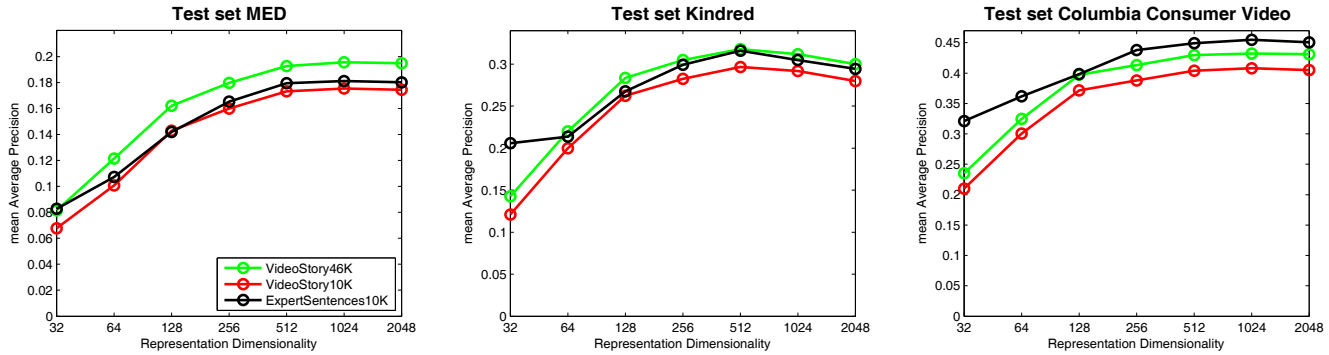

Figure 6: Description quality and quantity. As expected, the more reliable the correspondence between description and video content, the better the result. The event recognition accuracy obtained by ExpertSentences10K can be approached, and even improved, by simply harvesting more descriptions from the web.

correlation in the descriptions but are visually dissimilar. Combining these two terms together, as is done by Description embedding, undermines the accuracy of the classifiers predicting them from videos. In contrast, our VideoStory learns the embedding by taking both the term correlations and their visual similarity into account. In other words, in a VideoStory the correlated terms are combined only if their combination improves their classifier prediction. It prevents the combination of correlated terms which are visually dissimilar.

5.2 Description Quality and Quantity

The results of experiment 2 are presented in Figure 6. As expected, the more reliable the correspondence between the description and the video content it describes, the better the result. A VideoStory learned from ExpertSentences10K always performs better than an embedding learned from the same amount of descriptions from VideoStory10K. Yet it should be noted that the expert descriptions are generally unavailable or hard to obtain. Interestingly, the event recognition accuracy obtained by expert descriptions can be approached, and even improved (for both the test set MED and test set Kindred) by simply harvesting more descriptions from the web. When considering all results, the complete VideoStory46K dataset is the best choice overall. The embedding reduces the influence of noisy video descriptions, especially when the number of input videos and descriptions are large. It demonstrates the value of user generated videos and descriptions as an unlimited, free, yet precious resource for constructing an effective VideoStory.

5.3 VideoStory vs Others

Table 2 and Table 3 show that the VideoStory outperforms all three state-of-the-art video representations on all three test sets. By comparing the VideoStory and the low-level representation we observe a higher event recognition accuracy of 0.196 vs 0.174 for the MED test, 0.312 vs 0.263 for the Kindred test, and 0.432 vs 0.409 for the Columbia Consumer Video test set. It demonstrates that the VideoStory enriches the video representation by transferring and incorporating semantics from the descriptions.

The results further demonstrate that the VideoStory outperforms the attributes and the informative attributes with ease. We provide two reasons to explain this. First, both the attributes and the informative attributes rely on the TRECVID Semantic Indexing task and the ImageNet categories as attributes. However, many of these pre-specified

Table 3: VideoStory vs Others. VideoStory outperforms state-of-the-art video representations on the Columbia Consumer Videos dataset.

	Test set Columbia Consumer Video			
Event	Attributes [11]	Informative [24]	Low-Level [37]	VideoStory
Basketball	0.293	0.317	0.485	**0.553**
Baseball	0.401	**0.463**	0.298	0.299
Soccer	0.336	0.302	0.469	**0.505**
Ice skating	0.632	0.649	0.646	**0.675**
Skiing	0.641	0.651	0.610	**0.671**
Swimming	0.520	0.489	0.691	**0.764**
Biking	0.324	0.307	0.420	**0.561**
Graduation	0.083	0.058	**0.135**	0.121
Birthday	0.149	0.216	0.187	**0.257**
Wedding reception	0.147	**0.201**	0.124	0.117
Wedding ceremony	0.216	0.248	**0.387**	0.324
Wedding dance	0.243	0.294	**0.550**	0.521
Music performance	**0.279**	0.247	0.225	0.201
Non-music performance	0.195	0.190	**0.334**	0.282
Parade	0.247	0.295	0.579	**0.634**
mean	0.314	0.328	0.409	**0.432**

attributes are not semantically relevant for the events. For example, a considerable number of ImageNet categories are devoted to specific animal species, which are not characteristic for representing events. Although the informative attributes address this problem by selecting only the most relevant subset of attributes per event, their performance is limited to the availability of relevant attributes. If the relevant attributes are unavailable in the initial pool, they can never be selected. In contrast to pre-specified attributes, the VideoStory embedding is automatically derived from the VideoStory46K dataset, which includes many descriptions relevant to events. Second, 1,000 out of 1,346 pre-specified attributes are derived from ImageNet images, for which we can only extract static visual features. As a consequence, all the pre-specified attribute representations, which rely on image data to train attribute predictors, can not benefit from the state-of-the-art motion features for event recognition. Hence, it limits their performance in comparison to our VideoStory which is trained from *video* data. The results demonstrate the importance of learning a representation from event descriptions and their corresponding video data, which are both achieved by using the VideoStory.

Apart from their better event recognition accuracy, the VideoStory is also extracted much more efficiently compared to the attribute based representation. The attribute based representation is trained on images and so can only be applied to video frames. Hence, extracting the attribute-based

Table 2: VideoStory vs Others. VideoStory outperforms the state-of-the-art on MED and Kindred test set.

	Test set MED				Test set Kindred			
Event	Attributes [11]	Informative [24]	Low-Level [37]	VideoStory	Attributes [11]	Informative [24]	Low-Level [37]	VideoStory
Birthday party	0.089	0.103	0.083	**0.118**	0.365	**0.379**	0.224	0.331
Changing vehicle tire	0.217	**0.239**	0.106	0.103	0.087	0.109	0.167	**0.180**
Flash mob gathering	0.432	0.434	**0.544**	0.535	0.078	0.080	0.248	**0.309**
Getting vehicle unstuck	0.307	0.309	0.137	**0.319**	0.354	0.371	0.301	**0.393**
Grooming animal	0.102	0.110	0.114	**0.151**	0.328	0.336	0.381	**0.501**
Making sandwich	0.055	0.054	0.073	**0.074**	0.297	0.296	**0.356**	0.278
Parade	0.195	0.198	0.352	**0.452**	0.056	0.059	0.106	**0.146**
Parkour	0.170	0.184	0.705	**0.721**	0.023	0.037	0.619	**0.792**
Repairing appliance	0.143	0.163	0.174	**0.184**	0.111	0.131	**0.540**	0.534
Working sewing project	0.081	0.106	0.085	**0.151**	0.022	0.047	0.327	**0.488**
Attempting bike trick	0.135	**0.144**	0.033	0.061	0.042	0.041	0.099	**0.198**
Cleaning appliance	0.007	0.033	0.072	**0.078**	0.008	0.034	0.110	**0.162**
Dog show	0.164	0.187	**0.409**	0.354	0.133	0.156	**0.479**	0.416
Giving directions location	0.007	0.018	**0.047**	0.004	0.003	**0.014**	0.004	0.003
Marriage proposal	0.002	**0.018**	0.007	0.004	0.003	**0.019**	0.008	0.008
Renovating home	0.047	0.047	**0.072**	0.051	0.141	**0.142**	0.112	0.131
Rock climbing	0.090	0.101	**0.118**	0.100	0.244	0.255	0.557	**0.618**
Town hall meeting	0.157	**0.176**	0.149	0.118	0.097	**0.116**	0.065	0.061
Winning race without vehicle	0.206	0.210	0.130	**0.217**	0.243	0.255	0.308	**0.413**
Working metal crafts project	0.090	0.101	0.068	**0.118**	0.083	0.104	0.241	**0.278**
mean	0.135	0.147	0.174	**0.196**	0.136	0.149	0.263	**0.312**

representation for video includes many predictions for each individual frame, before they are aggregated per video. In contrast, our proposed VideoStories are directly trained on videos and thus provide video-level predictions, which makes it significantly more efficient.

Finally, we evaluate the video descriptions generated by VideoStory and by the attribute baseline [11]. Following the protocol of [7], we use the ROUGE-1 metric on the automatically generated descriptions, with the expert-provided descriptions as ground truth. For the generated descriptions we predefine the length k, and use the highest scoring terms from Eq. (8) for VideoStory, see Figure 8 for some examples of videos and their predicted terms. As baseline, we use the highest scoring attributes names.

In Figure 7 we show the results on the MED test and Kindred test sets, where expert-provided descriptions are available for each video. Since ROUGE-1 is a recall based metric, it computes the recall of the ground truth terms in the provided description, we evaluate different values of k. From the results we observe that VideoStory generates more accurate video translations, and the performance gap increases for a higher value of k. This could be explained by the fact that the attribute baseline uses only terms of TRECVID and ImageNet categories, many being very specific and rarely used by humans to describe videos. In contrast, VideoStory relies on human-provided descriptions obtained from the web, which leads to a video description that is more in-line with the ground truth descriptions.

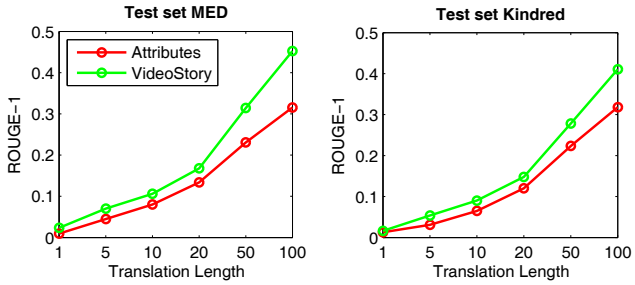

Figure 7: VideoStory vs Others. VideoStory generates more accurate descriptions.

6. CONCLUSION

In this paper we propose *VideoStory* a new multimedia embedding for few-example event recognition and translation. Our joint objective function aims to optimize the textual descriptiveness as well as the visual predictability of the embedding. In contrast to previous approaches, VideoStory allows both to train event classifiers using just a few examples in a low-dimensional, yet highly-discriminative, embedding space and to translate embedded videos to text. For training our VideoStory embedding, we rely on the weak supervision of easy to harvest descriptions from web videos. Therefore, we have introduced the VideoStory46K dataset, consisting of 46K YouTube videos with their title captions as descriptions. Results on three challenging test sets show that our event classification framework outperforms the current state-of-the-art. Moreover, we are able to generate human interpretable translations for previously unseen videos, opening up new connections with natural language processing and computational linguistics for describing and querying videos.

Acknowledgments This research is supported by the STW STORY project, the Dutch national program COMMIT, and by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20067. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

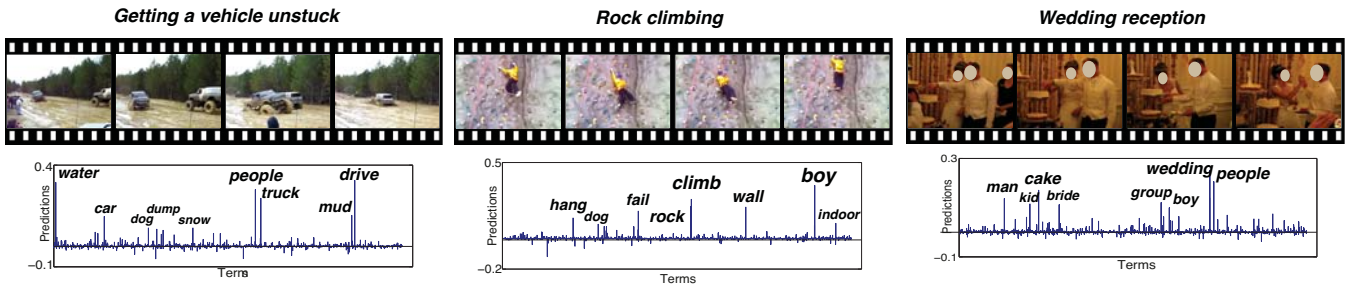

Figure 8: VideoStory event recognition and translation results on previously unseen videos.

7. REFERENCES

[1] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In *CVPR*, 2013.

[2] R. Aly et al. The AXES submissions at trecvid 2013. In *TRECVID*, 2013.

[3] S. Banerjee and T. Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *IJCAI*, 2003.

[4] T. Berg, A. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *ECCV*, 2010.

[5] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *ICCS*, 2010.

[6] Q. Chen et al. Spatio-temporal fisher vector coding for surveillance event detection. In *ACM MM*, 2013.

[7] P. Das, R. Srihari, and J. Corso. Translating related words to videos and back through latent topics. In *WSDM*, 2013.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[9] J. Dodge et al. Detecting visual text. In *NAACL*, 2012.

[10] S. Guadarrama et al. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *ICCV*, 2013.

[11] A. Habibian and C. Snoek. Recommendations for recognizing video events by concept vocabularies. *CVIU*, 124, 2014.

[12] T. Hofmann. Probabilistic latent semantic indexing. In *ACM SIGIR*, 1999.

[13] H. Izadinia and M. Shah. Recognizing complex events using large margin joint low-level event model. In *ECCV*, 2012.

[14] M. Jain, H. Jégou, and P. Bouthemy. Better exploiting motion for better action recognition. In *CVPR*, 2013.

[15] Y.-G. Jiang, G. Ye, S.-F. Chang, D. Ellis, and A. Loui. Consumer video understanding: A benchmark database and an evaluation of human and machine performance. In *ICMR*, 2011.

[16] Y.-G. Jiang, X. Zeng, G. Ye, S. Bhattacharya, D. Ellis, M. Shah, and S.-F. Chang. Columbia-ucf trecvid2010 multimedia event detection: Combining multiple modalities, contextual concepts, and temporal matching. In *TRECVID*, 2010.

[17] L. Kennedy, S.-F. Chang, and I. Kozintsev. To search or to label?: predicting the performance of search-based automatic image classifiers. In *ACM MIR*, 2006.

[18] D. Klein and C. Manning. Accurate unlexicalized parsing. In *ACL*, 2003.

[19] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[20] P. Kuznetsova, V. Ordonez, A. Berg, T. Berg, and Y. Choi. Generalizing image captions for image-text parallel corpus. In *ACL*, 2013.

[21] J. Liu et al. Video event recognition using concept attributes. In *WACV*, 2013.

[22] Z. Ma, Y. Yang, Y. Cai, N. Sebe, and A. Hauptmann. Knowledge adaptation for ad hoc multimedia event detection with few exemplars. In *ACM MM*, 2012.

[23] Z. Ma, Y. Yang, Z. Xu, N. Sebe, and A. Hauptmann. We are not equally negative: fine-grained labeling for multimedia event detection. In *ACM MM*, 2013.

[24] M. Mazloom, E. Gavves, K. van de Sande, and C. Snoek. Searching informative concept banks for video event detection. In *ICMR*, 2013.

[25] T. Mei, Y. Rui, S. Li, and Q. Tian. Multimedia search reranking: A literature survey. *ACM CS*, 46(3), 2014.

[26] M. Merler, B. Huang, L. Xie, G. Hua, and A. Natsev. Semantic model vectors for complex video event recognition. *IEEE TMM*, 14(1), 2012.

[27] P. Natarajan et al. Multimodal feature fusion for robust event detection in web videos. In *CVPR*, 2012.

[28] A. Natsev, M. Naphade, and J. Tešić. Learning the semantics of multimedia queries and concepts from a small number of examples. In *ACM MM*, 2005.

[29] P. Over, J. Fiscus, G. Sanders, et al. TRECVID 2013-an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *TRECVID*, 2013.

[30] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE TPAMI*, 27(8), 2005.

[31] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. Lanckriet, R. Levy, and N. Vasconcelos. A new approach to cross-modal multimedia retrieval. In *ACM MM*, 2010.

[32] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *IJCV*, 105(3), 2013.

[33] S. Strassel et al. Creating havic: Heterogeneous audio visual internet collection. In *LREC*, 2012.

[34] A. Tamrakar et al. Evaluation of low-level features and their combinations for complex event detection in open source videos. In *CVPR*, 2012.

[35] A. Ulges, C. Schulze, M. Koch, and T. Breuel. Learning automatic concept detectors from online video. *CVIU*, 114(4), 2010.

[36] K. van de Sande, T. Gevers, and C. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE TPAMI*, 32(9), 2010.

[37] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[38] Q. Wang, J. Xu, H. Li, and N. Craswell. Regularized latent semantic indexing. In *ACM SIGIR*, 2011.

[39] X.-J. Wang, L. Zhang, X. Li, and W.-Y. Ma. Annotating images by mining image search results. *IEEE TPAMI*, 30(11), 2008.

[40] J. Weston, S. Bengio, and N. Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *IJCAI*, 2011.

[41] Y. Yang and M. Shah. Complex events detection using data-driven concepts. In *ECCV*, 2012.