University of Amsterdam and Renmin University at TRECVID 2017: Searching Video, Detecting Events and Describing Video

Cees G. M. Snoek*, Xirong Li†, Chaoxi Xu†, Dennis C. Koelma*

*University of Amsterdam
Amsterdam, The Netherlands

†Renmin University of China
Beijing, China

Abstract

In this paper we summarize our TRECVID 2017 [1] video recognition and retrieval experiments. We participated in three tasks: video search, event detection and video description. For both video search and event detection we explore semantic representations based on VideoStory [8] and an ImageNet Shuffle [16], which thrive well in few-example regimes. For the video description task we experiment with a deep network that predicts a visual representation from a natural language description with Word2VisualVec [5], and use this space for the sentence matching. For generative description we enhance a neural image captioning model with Early Embedding and Late Reranking [4]. The 2017 edition of the TRECVID benchmark has been a fruitful participation for our joint-team, resulting in the best overall result for video search and event detection as well as the runner-up position for video description.

1 Video Search and Event Detection

The MediaMill approach to ad hoc video search and 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 [15]. 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 [8, 9]. 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:

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

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 $\boldsymbol{S}$:

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

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.* $\Omega(\boldsymbol{A}) = \frac{1}{2}\|\boldsymbol{A}\|_{\mathrm{F}}^2 = \frac{1}{2}\sum_i \|\boldsymbol{a}_i\|_2^2 = \frac{1}{2}\sum_{ij} a_{ij}^2$, 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:

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

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 pre-trained 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 [17] 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 [16]. The representation starts from a deep network 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 [16]. 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

We consider two base networks for both VideoStory and ImageNet Shuffle: ResNet [10] and ResNeXt [20]. For VideoStory, we consider four training regimes: i) the first one relies on the VideoStory46K dataset [8], ii) the second trains on the Fudan Columbia Video dataset (FCVID) [11], iii) the third trains on EventNet [22], and iv) trains on a merged collection of all three. We only add terms to the dictonary that occur more than 10 times.

1.3.1 Ad Hoc Video Search

Our ad hoc video search experiments show it is important to select the right terms. Instead of just taking the average of the query terms in word2vec space, we consider a query plan based on part-of-speech tagging of the query. We consider the following tags: $< noun1 >$, $< verb >$, $< noun2 >$, $< subject >$, $< predicate >$, $< remainder >$. Our query plan is as follows: A) We use nouns, verbs, and adjectives in $< subject >$ unless it concerns a person (noun1 = "person", "man", "woman", "child", ...). B) We use nouns in $< remainder >$ unless it concerns a person or noun is a setting ("indoors", "outdoors", ...). C) We use $< predicate >$, and D) We use all nouns in the sentence, unless noun is a person or a setting.

We further find that the ResNeXt base network is better than ResNet for VideoStory, while it is the other way around for the ImageNet Shuffle embedding. Overall, the VideoStory embedding (Run3) slightly outperforms the ImageNet Shuffle (Run4). A Borda count combination of multiple Shuffle embeddings results in better retrieval than a combination of VideoStory embeddings, and we obtain the best results when fusing the VideoStory and ImageNet Shuffle retrieval results (Run2 and Run1). Our first three runs are the top-3 submissions for the automatic ad hoc video search task in the TRECVID 2017 benchmark, see Figure 1.

1.3.2 Multimedia Event Detection

Our multimedia event detection experiments follow our winning entry of the TRECVID 2016 benchmark [18]. We rely on the VideoStory and ImageNet Shuffle embedding, as well as dense trajectories and MFCC audio features. New this year are the updated base networks, difference coding instead of average pooling, and a zero-shot sliding window based augmentation. The difference coding performs a K-means clustering on the last fully connected layer before the probability layers, then it performs a Fisher like encoding but sigma is based on distance of points assigned to a cluster center. The zero-shot sliding window augmentation expands the ten training examples by finding segments in each training video most similar to the event query. We use a 30 second non-overlapping fixed window to segment the videos and rely on the cosine similarity between query text and VideoStory embedding. We select a maximum of five additional segments per training video.

We submit the following runs:

- Run5 computes the last fully connected layer of a ResNeXt Shuffle applied to two frames per second. A video is represented by difference coding of the frame level features. A HIK SVM model is trained on the difference codings and used to classify videos.
- Run4 adds HIK SVM models trained on VideoStory features based on a ResNeXt base network and additional examples as selected with the segment augmentation. The final output of the system is based on fusion of all three modalities.

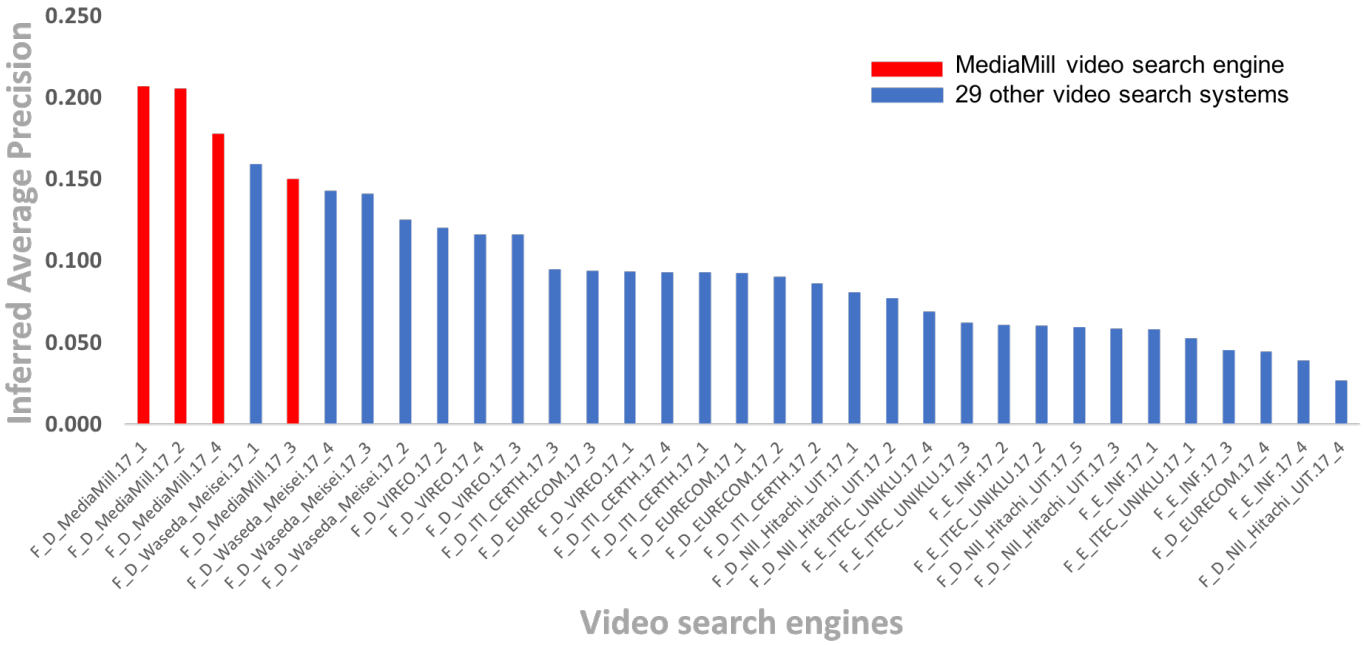

Figure 1: Overview of the 2017 TRECVID ad hoc video search task benchmark in which MediaMill was the best overall performer, all runs ranked according to mean inferred average precision.

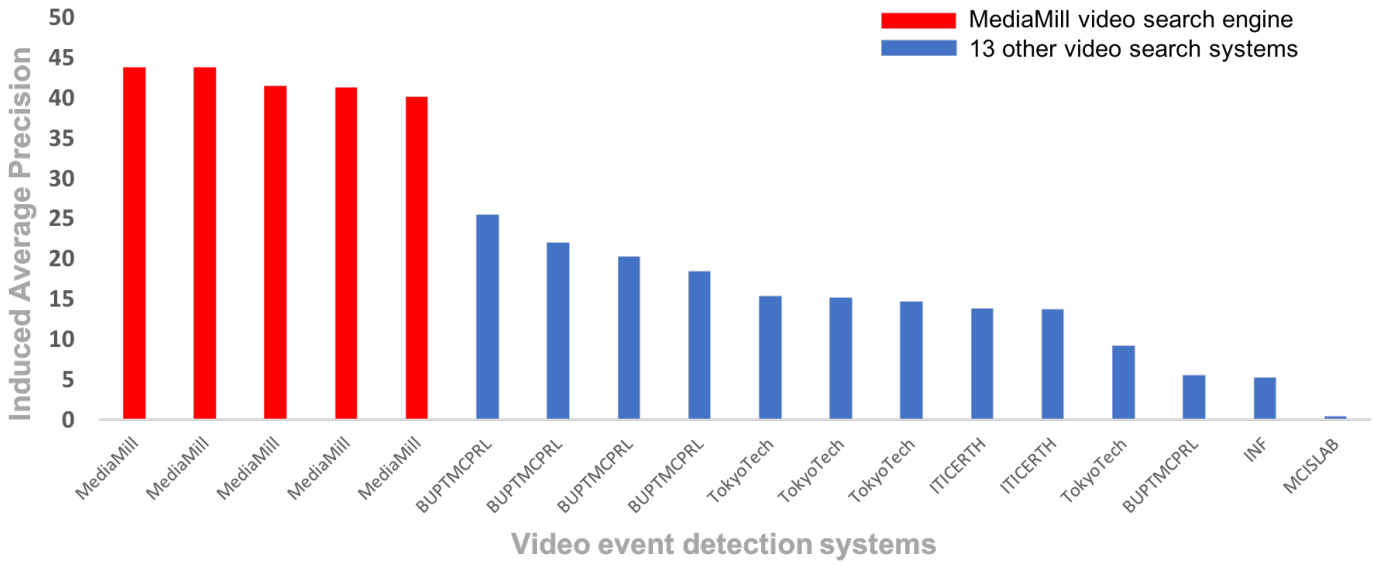

Figure 2: Overview of the 2017 TRECVID multimedia event detection task benchmark for pre-specified events in which MediaMill was the best overall performer, all runs ranked according to mean induced average precision.

- Run3 is similar to Run4, but also includes ResNet models. The final output of the system is based on fusion of all six modalities. For this system we also submitted an ad hoc run.

- Run2 is similar to Run4, but also adds HIK SVM models trained on Fisher vector encoded dense trajectory features and MFCC audio features. The final output of the system is based on fusion of all five modalities.

- Run1 is a combination of Run2 and Run3. The final output of the system is based on fusion of all eight modalities. For this system we also submitted an ad hoc run.

Run5 serves as a baseline and shows the value of difference coding with a mean iAP of 40.2, better than all other non-MediaMill runs submitted to TRECVID 2017. When combined with VideoStory and additional examples we score a

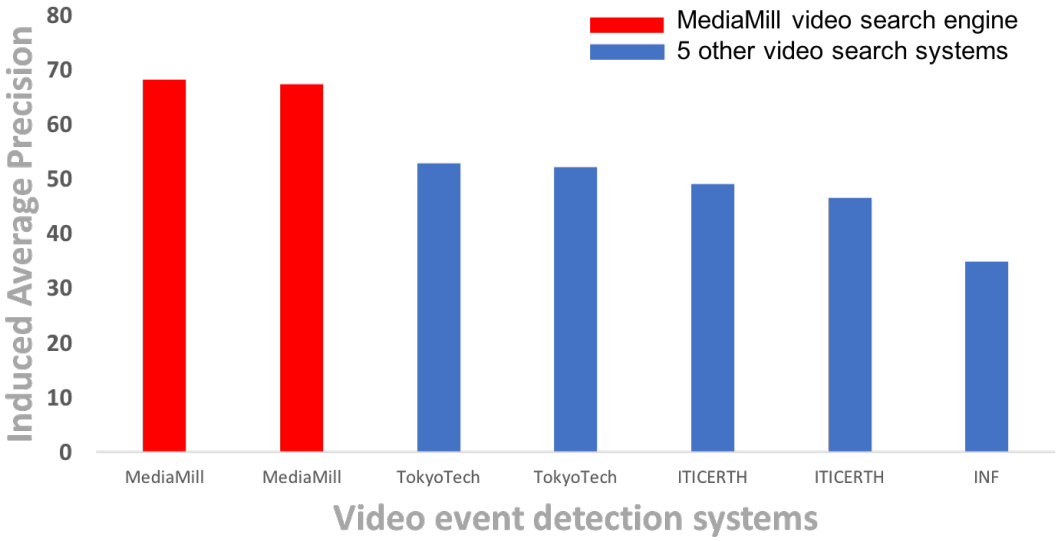

Figure 3: Overview of the 2017 TRECVID multimedia event detection task benchmark for ad hoc events in which MediaMill was the best overall performer, all runs ranked according to mean induced average precision.

mean iAP of 41.3. Adding an additional base network further lifts the iAP to 43.8. The addition of motion and audio has a modest effect overall (Run2: 41.5 / Run1: 43.8), but may still be beneficial for individual events. For event E060 "Cheerleading" for example, Run1 scores 73.0 where Run5 achieves 55.8. For the ad hoc runs, we see similar behavior, be it that with 68.2 the visual-only run (Run3) slightly outperforms the multimodal run scoring 67.3 (Run1). For both the pre-specified and ad hoc events, our runs are the top submissions for the multimedia event detection task in the TRECVID 2017 benchmark, see Figure 2 and Figure 3.

2 Video Description

We continued our participation in the Video to Text Description (VTT) task, which consists of two subtasks, *i.e.*, Matching and Ranking, and Description Generation. For both subtasks our system came in the second place.

2.1 Matching and Ranking

In this subtask, participants were asked to rank a list of pre-specified sentences in terms of their cross-modal relevance with respect to a given video. There are 1,880 videos collected from Twitter Vine for testing. Each video is about 6 sec long. The videos have been (randomly) split by the task organizers into four subsets, *i.e.* Subset.2, Subset.3, Subset.4 and Subset.5, which contain 1,613, 795, 388 and 159 videos, respectively. Subset.2 has two equal-sized sets of sentences, each containing one ground-truth sentence for each video in Subset.2. In a similar manner, the sentence sets for Subset.3, Subset.4 and Subset.5 are constructed but the number of the sets become 3, 4, and 5.

Approach. We improve our TV16 system by substituting the Word2VisualVec (W2VV) model [5] for its multi-scale version [6]. W2VV is a deep neural network that projects a given sentence into a visual feature space, enabling cross-modal matching between video and text to be directly conducted in the visual feature space. The projection is performed in two steps. First, a varied-length sentence s is encoded into a fixed-length vector by a sentence vectorization layer. Then, the encoding result goes through two fully connected layers to produce a visual feature vector $r(s)$. Consequently, the cross-modal relevance between a given video v and the sentence is computed as the cosine similarity between $r(s)$ and the video feature $\phi(v)$. While the TV16 edition implements the sentence vectorization using only a pre-trained word2vec (w2v) [13], the new W2VV performs multi-scale sentence vectorization. As illustrated in Fig. 4, the sentence is vectorized in parallel by three vectorization strategies including Bag-of-Words (BoW), w2v and a Gated Recurrent Units (GRU) [2].

To obtain the video-level feature, we uniformly sample frames with an interval of 0.5 second. A pre-trained CNN model is used to extract deep visual features per frame. The video-level feature is obtained by mean pooling over the frames. We employ two ResNeXt-101 models, separately trained on the full ImageNet dataset [3] and the Places2 scene dataset [23], and thus denoted as ResNeXt-101-imagenet and ResNeXt-101-places. The audio channel of a video can sometimes provide complementary information to the visual channel. To exploit this channel, we extract a 1,024-dim bag of quantized Mel-frequency Cepstral Coefficients vector [7] and concatenate it with the visual features. We train W2VV models that predict the visual-audio features using the MSR-VTT dataset [21]. The TV16 test set is adopted for optimizing hyper-parameters.

Table 1: Performance of our submissions in the TV17 video matching and ranking subtask.

2*runs	Subset.2			Subset.3				Subset.4					Subset.5					
	A	B	*MEAN*	A	B	C	*MEAN*	A	B	C	D	*MEAN*	A	B	C	D	E	*MEAN*
run 1	0.223	0.226	0.225	0.303	0.306	0.304	0.304	0.401	0.387	0.398	0.395	0.395	0.517	0.548	**0.586**	0.514	0.531	0.539
run 2	0.225	0.227	0.226	0.309	0.308	0.306	0.308	0.406	**0.392**	0.417	0.400	0.404	**0.532**	**0.561**	0.585	0.513	0.547	**0.548**
run 3	0.218	0.225	0.222	0.303	0.306	0.307	0.305	**0.407**	0.384	0.416	0.398	0.401	0.523	0.557	0.576	**0.528**	0.532	0.543
run 4	**0.229**	**0.229**	**0.229**	**0.316**	**0.312**	**0.310**	**0.313**	**0.407**	0.388	**0.421**	**0.404**	**0.405**	0.528	0.555	0.585	0.513	**0.548**	0.546

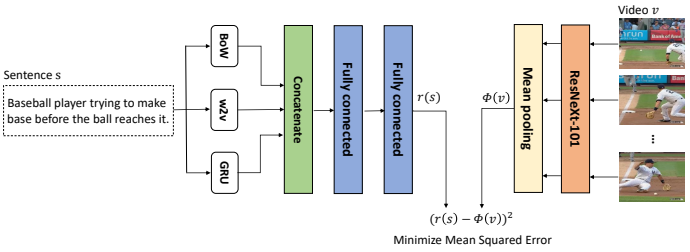

Figure 4: A conceptual diagram of the multi-scale Word2VisualVec model [6] we use for the Matching and Ranking subtask. The model transforms a given sentence into a video feature vector. The sentence is first vectorized in parallel by three vectorization strategies, *i.e.* Bag-of-Words (BoW), a pre-trained word2vec (w2v), and a Gated Recurrent Units (GRU) network. It then goes through two fully connected layers. The model is trained using many relevant video-sentence pairs to minimize the Mean Squared Error between the feature vectors extracted from the videos and the vectors predicted from their sentence descriptions.

Submissions. We submitted four runs:

- *run 1. baseline.* We use W2VV that predicts the ResNext-101-imagenet + MFCC feature.
- *run 2. score fusion.* We use two W2VV models, one for predicting the ResNext-101-imagenet + MFCC feature and the other for predicting the ResNext-101-places + MFCC feature. Accordingly, a video-sentence pair has two relevance scores, which are averaged.
- *run 3. rank fusion.* We convert the individual scores from *run 2* to ranks and aggregate the ranks.
- *run 4. score fusion + rerank.* We perform video tagging, and re-order the sentence ranking from *run 2* by matching the sentences with the predicted tags. We collect the top tags predicted by ResNext-101-imagenet, ResNext-101-fcvid trained on the FCVID dataset [11], and the neighbor voting algorithm [14] with the neighbors retrieved from the MSR-VTT dataset.

Results. The performance of the four runs, measured by Mean Inverted Rank, is summarized in Table 1. Among them *run 4* tops the performance.

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 in the Matching and Ranking subtask.

Approach. Similar to our TV16 system [18], we employ the Show and Tell model [19], which uses a ConvNet + LSTM framework for sentence generation. We use the implementation[1] of [12] that introduces importance sampling into the ConvNet + LSTM framework to regularize the influence of individual training captions in terms of their quality. We again use the MSR-VTT dataset [21] as our training data. We improve the generated description by reranking the sentences generated by the LSTM module in terms of automatically predicted video tags [4]. Moreover, to describe the "where" facet of a test video, we enrich the generated description by appending a scene phrase, e.g., on an airfield, in an art gallery, in a science museum, predicted by the ResNeXt-places model, if applicable.

Submissions. We submitted four runs, illustrated in Fig. 5 and descried as follows.

- *run 1. baseline.* We use the ResNeXt-101-imagenet + MFCC feature as input to the Show and Tell model, and perform beam search with size of 5 for sentence generation.
- *run 2. rerank.* We increase the beam size to 20 to generate a list of 20 candidate sentences. The sentence that maximizes its match with the predicted tags (c.f. Section 2.1) is chosen as the final video caption.
- *run 3. rerank + scene.* We enrich the result from *run 2* by appending a scene class predicted by ResNeXt-101-places, if applicable. To make the expanded sentence read naturally, we have compiled the 365 Place2 classes with prepositions and articles to proper phrases in advance. For instance, the class *indoor_stage* is replaced by "on an indoor stage".
- *run 4. rerank + scene + semantic input.* Based on *run 3*, we enrich the initial input to the LSTM network by concatenating the visual-audio feature and a 233-dim concept vector predicted by the ResNeXt-101-fcvid model.

Results. Table 2 summarizes the performance of the four runs on the TV17 test set. Sentence reranking by predicted tags gives better results under all metrics. The other tricks, *i.e. scene* and *semantic input*, do not really help for improving these automatically computed metrics.

[1] https://github.com/weiyuk/fluent-cap

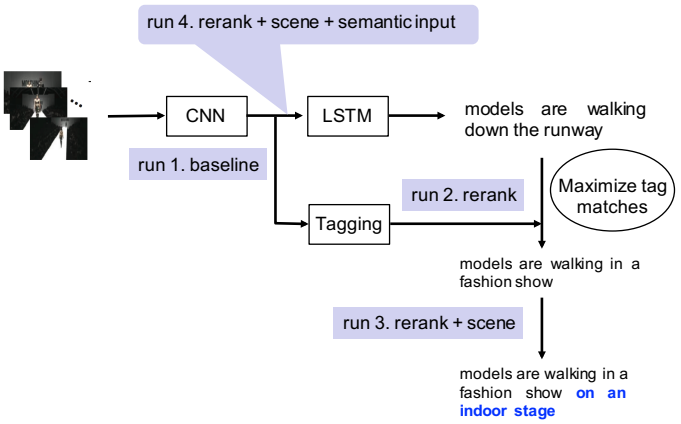

Figure 5: An illustration of our four runs for the Video Description Generation subtask. Among them *run 2* that reranks the generated sentences in terms of their matches with video tagging results performs the best on the TV17 test set.

Table 2: Performance of our submissions in the TV17 video description generation subtask.

Submissions	BLEU	METEOR	CIDEr	STS	SUM
run 1	0.013	0.152	0.291	0.418	0.875
run 2	**0.028**	0.181	**0.355**	**0.424**	**0.988**
run 3	0.020	**0.196**	0.328	0.401	0.945
run 4	0.024	0.194	0.328	0.402	0.947

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

[2] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, 2014.

[3] 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.

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

[5] 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.

[6] J. Dong, X. Li, and C. G. M. Snoek. Predicting visual features from text for image and video caption retrieval. *CoRR*, abs/1709.01362, 2017.

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

[8] 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.

[9] 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*, 39(10):2089–2103, 2017.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[11] 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. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(2):352–364, 2018.

[12] W. Lan, X. Li, and J. Dong. Fluency-guided cross-lingual image captioning. In *MM*, 2017.

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

[14] X. Li, C. G. M. Snoek, and M. Worring. Learning social tag relevance by neighbor voting. *IEEE Transactions on Multimedia*, 11(7):1310–1322, 2009.

[15] 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.

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

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

[18] 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. University of Amsterdam and Renmin University at TRECVID 2016: Searching video, detecting events and describing video. In *TRECVID Workshop*, 2016.

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

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

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

[22] G. Ye, Y. Li, H. Xu, D. Liu, and S.-F. Chang. Eventnet: A large scale structured concept library for complex event detection in video. In *MM*, 2015.

[23] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. In press.