DL-61-86 at TRECVID 2017: Video-to-Text Description

Jianfeng Dong[1,2], Shaoli Huang[1], Duanqing Xu[2], and Dacheng Tao[1]

[1]UBTECH Sydney Artificial Intelligence Centre, University of Sydney, Australia
[2]College of Computer Science and Technology, Zhejiang University, China

Abstract

In this paper, we summarize our work at the video-to-text description task (VTT) of TRECVID 2017. This year we participated in the matching and ranking subtask of VTT. Our entry is based on the Word2VisualVec [13] and a newly devised Spatial Enhanced Representation (SER). The Word2VisualVec is a deep neural network architecture that learns to predict a deep visual encoding of textual input. It is the winning entry in the VTT task of TRECVID 2016. We improve the Word2VisualVec by replacing the average pooling on the textual input with the multi-scale sentence vectorization [6] and using an improved triplet ranking loss [7]. The SER consists of two neural network branches which project videos and sentences into a learned latent space, respectively. For the video side branch, the model extracts an enhanced spatio-temporal representation for the input video. We implement this by learning a GRU with skip-connections that allow bypassing of the spatial feature. Our best run is the ensemble of six models which are variants of Word2VisualVec and SER. It leads the evaluation with a great margin in the context of all submissions from ten teams worldwide.

1. Approach

This year we participated in the matching and ranking subtask. In the subtask, participants were asked to rank a list of pre-defined sentences in terms of the cross-media similarity for a given video. This task is challenging, as videos and sentences are two distinct modalities and they are not directly comparable. Our solution is projecting the video and the sentence into a common space where their similarity is computed. We choose two spaces as the common space, that is, a visual feature space and learned latent space. For the visual feature space, we rely on Word2VisualVec, but using an improved triplet ranking loss [7]. The Word2VisualVec will not be introduced here. We refer the interested reader to [6]. For the learned latent space, we design a model that consists of two neural network branches to project sentences and videos into this space. On the sentence side branch, we utilize multi-scale sentence vectorization [6] to represent sentences. On the video side branch, we propose Spatial Enhanced Representation to extract an enhanced spatio-temporal representation from the input video. Hence, we name our proposed model as SER. In what follows, we introduce the input representation, followed by the details of the SER.

1.1. Input Representation

Video representation Following the good practice of using pre-trained ConvNets for video content analysis [5, 11, 15], we use a ResNet152 model [8] pre-trained on the full ImageNet dataset with over 10 million images and 10 thousand classes. Specifically, given a video, frames are uniformly sampled from the video with an interval of 0.5 seconds. For each frame, we take the output of the last pooling layer of the ResNet152 (pool5) as its feature vector. Finally, the input video is represented by $\{v_1, ..., v_N\}$, where N is the number of video frames and v_n is a 2048 dimensional feature vector corresponding to the n-th frame.

Sentence representation To handle sentences of varying length, we first vectorize each sentence. We employ multi-scale sentence vectorization [6] which jointly utilizes the bag-of-words (BoW), word2vec [12] and Gated Recurrent Unit (GRU) [4] based text encodings to vectorize the sentence. Given a sentence q, the representation is obtained by concatenating the three representations:

$$s(q) = bow(q) \Big|\Big| word2vec(q) \Big|\Big| gru(q), \qquad (1)$$

where $||$ denotes the concatenation operation of vectors, $bow(q)$, $word2vec(q)$ and $gru(q)$ respectively indicate BoW, word2vec and GRU based representation. For more details of multi-scale sentence vectorization, please refer to [6]. Note that the GRU is trained with the whole SER model in an end-to-end manner.

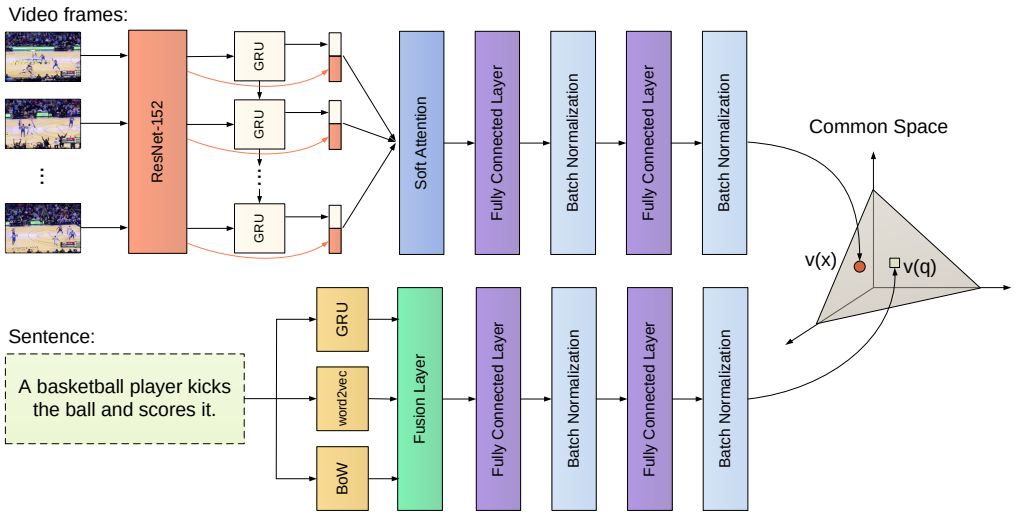


Figure 1. The overview of Spatial Enhanced Representation (SER) which project videos and sentence into a learned latent space by two neural network branches.

1.2. Spatial Enhanced Representation

Figure. 1 illustrates the structure of our proposed SER. It consists of two neural network branches that respectively project videos and sentences into a learned latent space. The similarity between videos and sentences can be readily computed in the learned latent space. We adopt the cosine similarity as the similarity metric. Given a video, we rank all candidate sentences in terms of their cosine similarity with the video.

Video side branch As videos contain rich spatial and temporal structure, capturing both spatial and temporal information is necessary. Average pooling and Recurrent Neural Network (RNN) are commonly used to encode videos. However, applying average pooling loses the temporal information contained in the video. Although using the RNN is able to capture more temporal information, it tends to lose some essential spatial information. Inspired by this, we propose to extract an enhanced spatio-temporal representation for videos. We implement this by learning a GRU with skip-connections that allow bypassing of the spatial feature. Moreover, a soft attention is employed on the output of the GRU to generate a context vector by attending to certain frames of the video.

More formally, given a video of $\{v_1, ..., v_N\}$, we first feed them sequentially into a GRU module to get a sequence of hidden state vectors $\{h_1, h_2, ..., h_N\}$,

$$h_n = GRU(v_n, h_{n-1}), n \in 1, 2, ..., N. \quad (2)$$

In order to prevent the loss of spatial information, we employ skip-connections to concatenate original visual features of video frames with the outputs of the GRU, obtaining a series of enhanced spatio-temporal feature vectors:

$$s_n = h_n || v_n, n \in 1, 2, ..., N. \quad (3)$$

Moreover, the soft attention mechanism is applied to the enhanced spatio-temporal feature vectors. For simplicity, we omit the bias term b in the following equations. The attention map $\alpha = \{\alpha_n\}_{n=1}^{N}$ is computed by a 2-layer neural network and the softmax function, that is:

$$m_n = tanh(W_s s_n) \odot tanh(W_v \overline{v}), n \in 1, 2, ..., N, \quad (4)$$

$$r_n = w_m m_n, n \in 1, 2, ..., N, \quad (5)$$

$$a = softmax(\mathop{||}_{n=1}^{N} r_n), \quad (6)$$

where $||$ denotes the concatenation operation of scalars, W_s, W_v are the trainable matrices, w_m is the trainable vector and $\odot$ indicates the element-wise multiplication. Additionally, $\overline{v}$ denotes the average feature vector of video frames, that is $\overline{v} = \frac{1}{N}\sum_{n=1}^{N} v_n$. With the attention map α, the attentive feature vector of the video is then calculated by

$$v_a = \sum_{n=1}^{N} \alpha_n s_n. \quad (7)$$

We further project the attentive feature vector v_a in a learned latent space by two fully connected layers. Each fully connected layer is followed by the batch normalization [9] and ReLU activations. More concretely, given a video x, we obtain the video representation $v(x)$ in the learned latent space as:

$$\begin{aligned} g_x &= \sigma(BN(W_{x'} v_a + b_{x'})), \\ v(x) &= \sigma(BN(W_{x''} g_x + b_{x''})), \end{aligned} \quad (8)$$

where $W_{x'}$ and $W_{x''}$ are the trainable parameters, $b_{x'}$ and $b_{x''}$ are the bias terms, BN indicates the batch normalization and $\sigma(\cdot)$ denotes the ReLU activation.

Sentence side branch Similar to the video video branch, two fully connected layers with batch normalization and ReLU activations are used to embed sentences into the learned latent space. Given a sentence q, the sentence is represented in the space as

$$\begin{aligned} q_x &= \sigma(BN(W_{q'} s(q) + b_{q'})), \\ v(q) &= \sigma(BN(W_{q''} g_q + b_{q''})), \end{aligned} \tag{9}$$

where $W_{q'}$ and $W_{q''}$ are the trainable parameters, $b_{q'}$ and $b_{q''}$ are the bias terms, and $s(q)$ indicates the multi-scale sentence feature vector computed in Eq. 1.

1.3. Learning algorithm

Objective function In order to train the model, we use an improved triplet ranking loss [7] which penalizes the model according to the hardest negative examples. Specially, the improved triplet ranking loss $l(x, q)$ for a video-sentence pair (x, q) is defined as:

$$l(x, q; \theta) = \max_{q'}[\alpha + s(x, q') - s(x, q)]_+, \tag{10}$$

where q' is a hardest negative sentence sample for the video-sentence pair (x, q), $[\cdot]_+$ indicates function of $[y]+ = max(y, 0)$, $s(\cdot, \cdot)$ denotes the cosine similarity function, and θ indicates all the trainable parameters in the model. Following [7], we define the hardest negative example as the most dissimilar sentence sample in a mini-batch. We train the model to minimize the overall improved triplet ranking loss on a given training set $\mathcal{D} = \{(x, q)\}$, containing a number of relevant video-sentence pairs:

$$\operatorname*{argmin}_{\theta} \sum_{(x,q)\in\mathcal{D}} l(x, q; \theta). \tag{11}$$

Optimization We solve Eq. (11) using stochastic gradient descent with RMSprop [14]. This optimization algorithm divides the learning rate by an exponentially decaying average of squared gradients, preventing the learning rate from effectively shrinking over time. We empirically set the initial learning rate $\eta = 0.0003$, decay weights $\gamma = 0.9$ and small constant $\epsilon = 10^{-6}$ for RMSprop. We apply the dropout to all hidden layers to mitigate model overfitting, with the rate of 0.2. Lastly, we take an empirical learning schedule as follows. Once the validation loss does not decrease in two consecutive epochs, we divide the learning rate by 2. The early stop occurs if the validation performance does not improve in five consecutive epochs. The maximal number of epochs is 50.

2. Evaluation

2.1. Dataset

This year NIST did not provide any training data, so we train our model on the aggregation of three external datasets, namely MSR-VTT [16], MSVD [3] and TGIF [10]. Additionally, we also use the training and the test set provided for the VTT task in TRECVID 2016 [1], as their video source are the same as the test set of this year. For the ease of reference, we name two datasets as tv2016train and tv2016test. We use the tv2016train for cross-validation and the tv2016test for fine-tuning the models trained on the external training sets. The tv2016train and the tv2016test respectively consist of 200 and 1,915 videos, and each video is associated with two sentences. An overview of datasets is given in Table 1. Note some videos of TGIF are corrupted, which have been removed. The TRECVID organizer provides four test sets this year, denoted as set 2, set 3, set 4 and set 5. Each test set has 1,613, 795, 388 and 159 videos respectively [2].

Table 1. Overview of datasets used in our submission.

	Dataset	# Videos	# Sentences
Train	MSVD [3]	1,970	80,863
	MSR-VTT [16]	10,000	200,000
	TGIF [10]	101,980	125,672
Validation	tv2016train [1]	200	400
Fine-Tune	tv2016test [1]	1,915	3,830

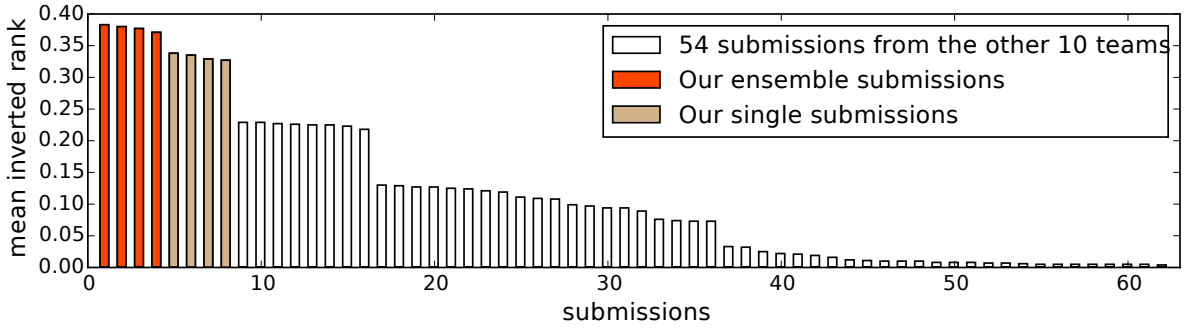
2.2. Submissions

DL-61-86.run1 Run1 is based on our entry for VTT task in TRECVID 2016 [13]. But we use new features extracted from ResNet152, use the multi-scale sentence vectorization to represent sentences and optimize the model with an improved triplet ranking loss. Moreover, we additionally apply the batch normalization after each fully connected layers, which further improve the performance. This run is not fine-tuned on the tv2016test set.

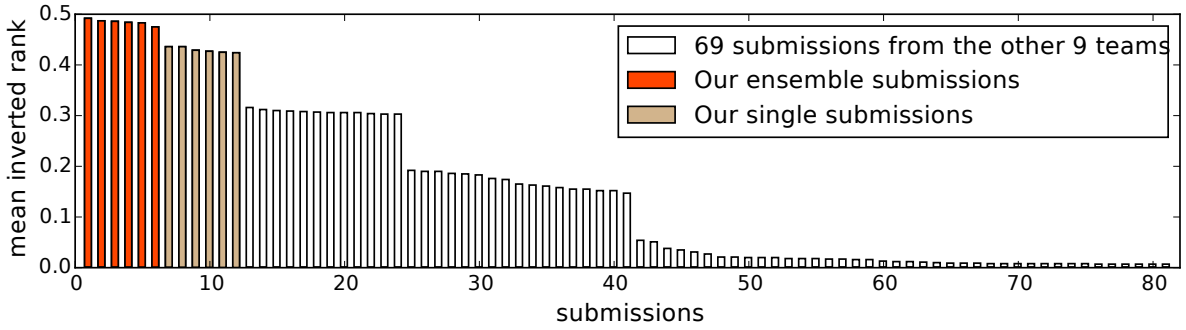
DL-61-86.run2 Run2 is the SER model described in the Section 1, which learns a learned latent space where the cosine similarity between the video and the sentence is computed. This run is also not fine-tuned on the tv2016test set.

DL-61-86.run3 Run3 ensembles six models: 1) run1, 2) Similar to run1, while it uses the word2vec as the sentence representation, 3) run2, 4) Similar to run2, while it uses the word2vec as the sentence representation, 5) Similar to run2, while it uses the average pooling instead of the soft attention, 6) Similar to run2, while it uses the word2vec as the sentence representation and the average pooling instead of the soft attention. Each model is trained with two different random initialization without fine-tuning. We fuse 12 similarities as the final similarity.

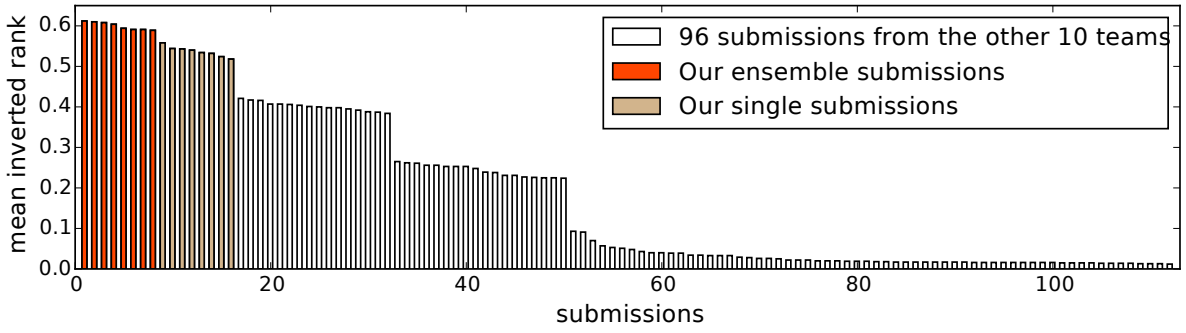
DL-61-86.run4 In run4, we ensemble six models used in run3, while we fine-tune them on the tv2016test set with initial learning rate 2e-5.



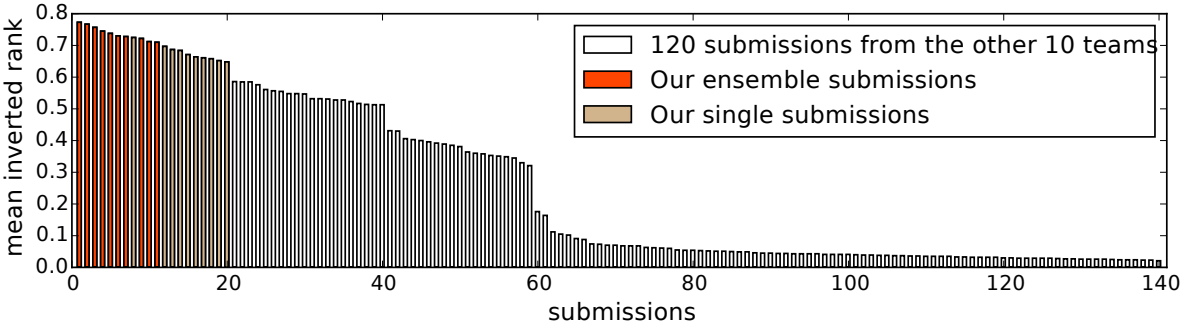
(a) Results on the test set 2



(b) Results on the test set 3



(c) Results on the test set 4



(d) Results on the test set 5

Figure 2. **State-of-the-art video-to-text matching and ranking results** in the TRECVID 2017 benchmark, showing the good performance of our models compared to other alternative approaches. The performance can be further improved by the model ensemble.

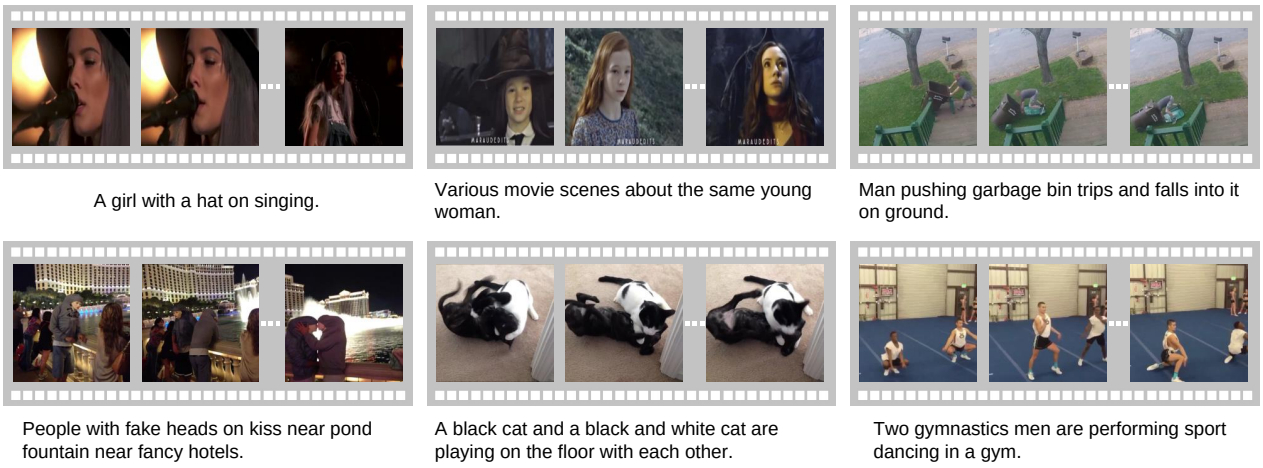


Figure 3. Results generated by our proposed SER model in the test set 2.

2.3. Results

The performance metric is Mean Inverted Rank at which the ground truth is found. Higher mean inverted rank means better performance. As shown in Figure. 2, our runs lead the evaluation with a great margin on all test sets. Moreover, model ensemble boosts the performance further, which demonstrates that the visual feature space and learned latent space are complementary for calculating the similarity. Some qualitative results generated by our proposed SER are shown in Figure. 3.

Acknowledgement

The authors are grateful to NIST and the TRECVID coordinators for the benchmark organization effort. This work was supported by the Australian Research Council Projects FL-170100117, DP-140102164, LP-150100671 and the key research development programs of Zhejiang province 2018C03051.

References

[1] G. Awad et al. Trecvid 2016: Evaluating video search, video event detection, localization, and hyperlinking. In *TRECVID*, 2016. 3

[2] G. Awad et al. Trecvid 2017: Evaluating ad-hoc and instance video search, events detection, video captioning and hyperlinking. In *TRECVID*, 2017. 3

[3] D. L. Chen and W. B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, 2011. 3

[4] 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. 1

[5] J. Dong, X. Li, W. Lan, Y. Huo, and C. G. Snoek. Early embedding and late reranking for video captioning. In *MM*, 2016. 1

[6] J. Dong, X. Li, and C. G. Snoek. Predicting visual features from text for image and video caption retrieval. *arXiv preprint arXiv:1709.01362*, 2017. 1

[7] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler. Vse++: Improved visual-semantic embeddings. *arXiv preprint arXiv:1707.05612*, 2017. 1, 3

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[9] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 2

[10] Y. Li, Y. Song, L. Cao, J. Tetreault, L. Goldberg, A. Jaimes, and J. Luo. TGIF: A New Dataset and Benchmark on Animated GIF Description. In *CVPR*, 2016. 3

[11] P. Mettes, D. Koelma, and C. Snoek. The ImageNet shuffle: Reorganized pre-training for video event detection. In *ICMR*, 2016. 1

[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013. 1

[13] C. G. Snoek, J. Dong, X. Li, X. Wang, Q. Wei, W. Lan, E. Gavves, N. Hussein, D. C. Koelma, A. W. Smeulders, et al. University of amsterdam and renmin university at trecvid 2016: Searching video, detecting events and describing video. In *Proceedings of the 14th TRECVID Workshop*, 2015. 1, 3

[14] 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. 3

[15] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko. Translating videos to natural language using deep recurrent neural networks. In *NAACL-HLT*, 2015. 1

[16] J. Xu, T. Mei, T. Yao, and Y. Rui. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*, 2016. 3