Attention Mechanisms, Signal Encodings and Fusion Strategies for Improved Ad-hoc Video Search with Dual Encoding Networks

Damianos Galanopoulos
CERTH-ITI
Thermi-Thessaloniki, Greece
dgalanop@iti.gr

Vasileios Mezaris
CERTH-ITI
Thermi-Thessaloniki, Greece
bmezaris@iti.gr

ABSTRACT

In this paper, the problem of unlabeled video retrieval using textual queries is addressed. We present an extended dual encoding network which makes use of more than one encodings of the visual and textual content, as well as two different attention mechanisms. The latter serve the purpose of highlighting temporal locations in every modality that can contribute more to effective retrieval. The different encodings of the visual and textual inputs, along with early/late fusion strategies, are examined for further improving performance. Experimental evaluations and comparisons with state-of-the-art methods document the merit of the proposed network.

CCS CONCEPTS

• **Computing methodologies**; • **Information systems** → **Video search**; **Query representation**;

KEYWORDS

Video search; Video retrieval; Ad-hoc video search; Deep learning; Dual encoding network; Attention mechanism

ACM Reference Format:
Damianos Galanopoulos and Vasileios Mezaris. 2020. Attention Mechanisms, Signal Encodings and Fusion Strategies for Improved Ad-hoc Video Search with Dual Encoding Networks. In *Proceedings of the 2020 International Conference on Multimedia Retrieval (ICMR '20), June 8–11, 2020, Dublin, Ireland.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3372278.3390737

1 INTRODUCTION

In the last years, the explosion of social media use has lead to a rapid increase in the multimedia content that is available on the Internet. This content originates from a variety of sources, and its nature is extremely heterogeneous, i.e. it includes video, images, audio, text etc., and combinations of them. Despite this multimodality, text-based queries remain the most natural way for people to search for content - be it video, images etc. The research field of text-based video retrieval, or more general cross-modal retrieval, addresses the problem of retrieving items of one modality (in our case, video) when the given query is of another modality (text).

A typical application scenario of text-based video search is Ad-hoc Video Search (AVS), originally introduced as a TRECVID benchmark task [23][1]. Given a set of unlabeled video shots and an unseen textual query, the goal of an AVS method is to retrieve the most related video shots, ranked from the most relevant to the least relevant shot for the query. The main challenge of AVS and its key difference from other video retrieval problems (e.g. concept-based retrieval [19][11]) is the lack of video examples for the queries. Moreover, these queries, which are given in natural language form, contain complex subject relations, e.g., *Find shots of exactly two men at a conference or meeting table talking in a room.*

Many methods have been proposed for the AVS problem in recent years, e.g. [19][15][25]. Their majority relies on examining the correlation of visual concepts with the textual queries, i.e. they use a variety of pre-trained visual concept detectors. The detectors' number and diversity are crucial for the retrieval performance.

During the last few years, several deep learning methods have been proposed for visual or text analysis and classification. Progress in the natural language processing field led to compelling text embedding methods [20][5] and gave the necessary boost to a variety of cross-model problems such as text-based video retrieval and image/video captioning. For this reason, recent AVS methods use deep learning for embedding the representation of different modalities (textual queries, videos) into a common subspace in a way that the new representations can be compared directly.

In this paper, we focus on the AVS problem. We use as a starting point a deep network architecture introduced in [8], in which two similar networks are jointly trained using several state-of-the-art (SoA) methods such as recurrent neural networks (GRUs) [4], text embeddings [20], and deep image classification networks [13][26]. We extend this by introducing two attention mechanisms. We also introduce and examine the impact of using more than one encodings of the visual content as well as of the textual query. As is typically the case in the relevant literature, pairs of video shots and captions are used for training the network. The main contributions of our work are summarized as follows:

- We integrate and evaluate two attention mechanisms into the dual encoding network. These lead to better textual and visual representation into the common subspace.
- We investigate the performance of different encodings for the text and the visual modalities.
- We compare early and late fusion for combining different encodings.

ICMR '20, June 8–11, 2020, Dublin, Ireland

ACM ISBN 978-1-4503-7087-5/20/06…$15.00
https://doi.org/10.1145/3372278.3390737

2 RELATED WORK

Early solutions to the AVS problem were based on large pools of visual concept detectors, and NLP techniques for query decomposition in order to identify concepts in the textual queries. In [19], a set of NLP rules and a variety of pre-trained deep neural networks for video annotation were used in order to associate visual concepts with the provided textual queries. In [14], a large amount of concept, scene and object detectors were used along with an inverted index structure for query-video association.

Recent SoA approaches rely on deep neural networks for directly comparing textual queries and the visual content in a common space [12]. Also, inspired by problems similar to AVS, e.g. cross-modal retrieval or visual question-answering, solutions that have been proposed for these problems were modified and adapted to AVS. In [9], an improved multi-modal embeddings system was proposed, together with a loss function that utilizes the hard negative samples of the dataset; this approach was adapted to the AVS problem in [3]. In [15], an improved version of the image-to-text matching method of [6] was proposed for the AVS task. More specifically, [15] used the method of [6] together with the triplet loss function of [9] and an improved sentence encoding strategy. In [22], a weakly-supervised method was proposed to learn a joint visual-text embedding space using an attention mechanism to highlight temporal locations in a video that are relevant to a textual description. This mechanism was also used for extracting text-depended visual features. Recently, the dual encoding network proposed in [8] encodes videos and queries into a dense representation using multi-level encodings for both text and videos and the improved loss function of [9]. In [10], the problem of video retrieval was addressed by training three different networks using different training datasets, and combining them by using an additional neural network.

3 PROPOSED METHOD

In this work, we propose an improved dual encoding method designed for Ad-hoc Video Search. Inspired by the dual encoding network presented in [8] (Section 3.1), we create a network that encodes video-caption pairs into a common feature subspace. In contrast to [8], our network utilizes attention mechanisms for more efficient textual and visual representation, and exploits the benefits of richer textual and visual embeddings.

Let **V** be a media item (e.g., an entire video or a video shot) and **S** the corresponding caption of **V**. Our network translates both **V** and **S** into a new common feature space $\Phi(\cdot)$, resulting in two new representations $\Phi(\mathbf{V})$ and $\Phi(\mathbf{S})$ that are directly comparable. For this, two similar modules, consisting of multiple levels of encoding, are utilized, for the visual and textual content respectively. Moreover, two new attention components are integrated into the baseline network. The overall network architecture is illustrated in Fig. 1.

3.1 Dual encoding network

For every video three different encodings are created, $\phi(V)^1$, $\phi(V)^2$, $\phi(V)^3$. We consider a video or a video shot as a sequence of n keyframes $\mathbf{V} = \{v_1, v_2, \dots, v_n\}$, where each keyframe vector v_i is the output of a pre-selected hidden layer of a pretrained deep network, e.g. the pool5 layer of Resnet [13] or Resnext [26]. The first encoding is the global representation of every video and is obtained by mean pooling the individual keyframe representations, as follows: $\phi(V)^1 = \frac{1}{n}\sum_{i=1}^{n} v_i$.

Next, the keyframe representation vectors $\{v_1, v_2, \dots, v_n\}$ are fed in a sequence of bi-directional Gated Recurrent Units [4] (bi-GRUs). The hidden state in time t of a forward $\overrightarrow{GRU}$ is defined as $\overrightarrow{h_t} = \overrightarrow{GRU}(v_t, \overrightarrow{h_{t-1}})$, and in the backward $\overleftarrow{GRU}$ as $\overleftarrow{h_t} = \overleftarrow{GRU}(v_t, \overleftarrow{h_{t+1}})$. All GRU's hidden states are represented as a feature matrix $\mathbf{H_v} = [h_1, h_2, \dots, h_n]$, where $h_t = concat(\overrightarrow{h_t}, \overleftarrow{h_t})$. To obtain the second level of encoding, mean pooling of the h_t values is performed as follows: $\phi(V)^2 = \frac{1}{n}\sum_{t=1}^{n} h_t$.

Subsequently, a 1-d CNN is built and fed with the feature matrix $\mathbf{H_v}$. A convolutional layer $Conv1d_{k,r}$ is used, with r filters of size k. After applying ReLU activation and max pooling to the layer's output, the $c_k = maxpool(ReLU(Conv1d_{k,r}(\mathbf{H_v})))$ vector is produced. Multiple representations of the video are created, using different $k = 2, 3, 4, 5$ values. The third-level video representation is the concatenation of the produced c_k vectors: $\phi(V)^3 = [c_2, c_3, c_4, c_5]$.

Finally, the concatenation of the previously generated features is used as the global and multi-level feature representation of a video:

$$\phi(V) = BN(W_v concatt(\phi(V)^1, \phi(V)^2, \phi(V)^3) + b_v)$$

where W_v and b_v are trainable parameters and BN a batch normalization layer.

Similar to the visual content encoding network, a multilevel encoding $\phi(S)^1$, $\phi(S)^2$, $\phi(S)^3$ is generated for the textual content. Given a sentence S containing m words, the $\phi(S)^1$ representation is created by averaging individual one-hot-vectors $\{w_1, w_2, \dots, w_m\}$. Next, as the second level of textual encoding, a deep network-based representation for every word is used as input for the bi-directional GRU module, and similarly to $\phi(V)^2$, $\phi(S)^2 = \frac{1}{m}\sum_{t=1}^{m} h_t$. Next, the feature matrix $\mathbf{H_s}$ of the textual bi-GRUs is forwarded into a 1-d convolutional layer with filter sizes $k = 2, 3, 4$ and $\phi(S)^3$ is calculated similarly to $\phi(V)^3$ above. The final textual representation is:

$$\phi(S) = BN(W_s concatt(\phi(S)^1, \phi(S)^2, \phi(S)^3) + b_s)$$

Following [9], [21] and [8], the improved marginal ranking loss is used to train the entire network.

3.2 Introducing Self-Attention Mechanisms

The 1-d CNN layer that is fed with $\mathbf{H_s}$ or $\mathbf{H_v}$ in the original network of [8] treats each item of the words or frames sequence equally. Our target is to exploit the most meaningful information from the textual and visual sequences, particularly the words with the highest semantic importance and the keyframes which are more representative for a video shot. For this, we introduce a self-attention mechanism [2][18] in each modality, in order to find the relevant importance of each word in the input sentence, and to find important temporal locations in a video-shot. An overview of this self-attention mechanism is illustrated in Fig. 2.

In the textual encoding part of the network, given the output of the bi-GRU $\mathbf{H_s}$, the attention model outputs a vector a:

$$a = softmax(w_{s2} tanh(W_{s1} \mathbf{H_s}^T))$$

where W_{s1} is a trainable weight matrix of size $d \times 2u$, where d is a hyper-parameter, $2u$ is the size of a single bi-GRU unit and w_{s2} a parameter vector of size d. The w_{s2} vector is extended in a

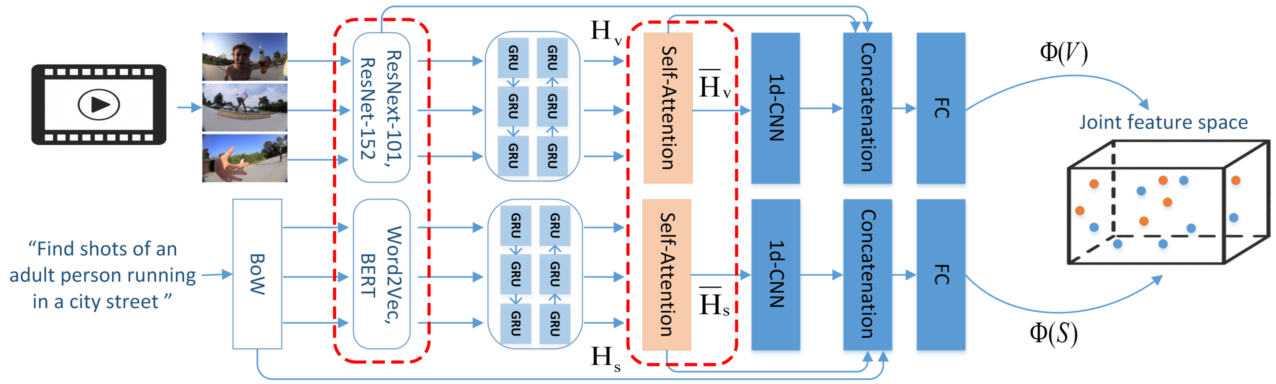

Figure 1: The overall dual encoding network incorporating the self-attention mechanism in both visual and textual modules. A pair of video+text is fed into the network in order to be represented into a joint feature space. The dotted red rectangles indicate the contributions of this work beyond [8]: the self-attention mechanisms, and the multiple video/text encodings.

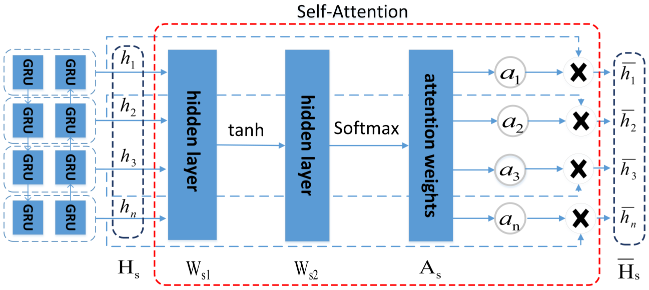

Figure 2: An illustration of the employed self-attention mechanism.

$z \times d$ matrix W_{s2} for multi-head attention, by modeling z semantic aspects of the text, as in [18], resulting in a weight matrix $\mathbf{A_s}$:

$$\mathbf{A_s} = softmax(W_{s2} tanh(W_{s1} \mathbf{H_s}^T))$$

The $softmax()$ is used for weight normalization, so that all the weights sum up to 1. Then, the attention matrix $\mathbf{A_s}$ is multiplied with the initial $\mathbf{H_s}$, resulting in matrix:

$$\overline{\mathbf{H}}_\mathbf{s} = \mathbf{A_s H_s}$$

$\overline{\mathbf{H}}_\mathbf{s}$ is forwarded into the 1-d convolutional layer instead of the feature matrix $\mathbf{H_s}$, as described in Sec. 3.1. This text-based self-attention mechanism is denoted as *Att* in the sequel.

A similar self-attention mechanism, denoted as *Atv*, is integrated in the visual encoding module. In this case $\mathbf{H_v}$ is used for calculating the attention weighted matrix $\mathbf{A_v}$, resulting in

$$\overline{\mathbf{H}}_\mathbf{v} = \mathbf{A_v H_v}$$

3.3 Examining multiple encodings and fusion strategies

Dealing with such a demanding task, where typically the SoA methods achieve accuracy of about $10 - 22\%$ on different evaluation datasets, it is vital to exploit the advantages of different signal encodings. Regarding the video module, two SoA deep neural network architectures are used for frame feature extraction: the ResNext-101 [26] and ResNet-152 [13] models. Concerning the text module, the performance of the Word2Vec [20] model, as well as the bidirectional transformer-based language model BERT [5], are examined. We also examine early fusion (as shown in Fig. 1) i.e. concatenation of encoding vectors, versus late fusion (i.e. merging of ranked lists, each obtained using a different text-visual encoding pair), for jointly exploiting the multiple encodings.

4 EXPERIMENTS

4.1 Experimental setup

We train our network[1] using the combination of two large-scale video datasets: MSR-VTT [27] and TGIF [17]. We evaluate its performance on the official evaluation dataset of the TRECVID AVS task for the years 2016, 2017, and 2018, i.e. the IACC.3 test collection consisting of 4,593 videos and altogether 335,944 shots. As evaluation measure we use mean extended inferred average precision (MXinfAP), which is an approximation of the mean average precision suitable for the partial ground-truth that accompanies the TRECVID dataset. As initial frame representations, generated by a ResNext-101 (trained on the ImageNet-13k dataset) and a ResNet-152 (trained on the ImageNet-11k dataset), we use the publicly-available features released by [15]. Also, two different word embeddings are utilized: i) the Word2Vec model [20] trained on the English tags of 30K Flickr images, provided by [7]; and, ii) the pre-trained language representation BERT [5], trained on Wikipedia content.

4.2 Results and discussion

For comparison reasons, we used the publicly available code of [8] to re-train the network with the same configuration and features we use in our methods. This method is indicated as *W2V+ResNext-101* in Table 1 and is used as a baseline for our experiments. Overall, three general network architectures are trained, i) the baseline network, ii) the network with the text-based attention mechanism, and iii) the network with the visual-based attention. Each network is trained using one or both available word embeddings (i.e., Word2Vec [a.k.a.

[1] Software available at: github.com/bmezaris/AVS_dual_encoding_attention_network

Table 1: Results (MXinfAP) of the proposed networks and their combinations, compared with the baseline [8]. The best results for each dataset are indicated with bold, while those that are worse than the baseline are given in parenthesis. All reported training/inference times are in hours, for a single setup (should be multiplied by 6 for the *Combination of 6 setups*) and for processing the whole training/test dataset. These numbers are not to be confused with query execution time; this is approximately 30 sec. for all but the late fusion methods, and 4 times higher for the latter.

		I. Combination of 6 setups			II. Best of 6 setups			Avg. training time	Inference time
		AVS16	AVS17	AVS18	AVS16	AVS17	AVS18		
(a)	W2V + ResNext-101 [8]	0.142	0.2189	0.1187	$0.1457^{\dagger,1}$	$0.212^{*,3}$	$0.1165^{*,2}$	6.66	1.72
(b)	(a) + *Att*	0.1544	0.2264	0.1233	$0.1477^{\dagger,2}$	$0.2298^{\dagger,1}$	$0.1183^{\dagger,2}$	7.53	1.81
(c)	(a) + *Atv*	0.1497	0.2274	0.1231	$0.1464^{\dagger,2}$	$0.2165^{\dagger,1}$	$0.1237^{\dagger,1}$	7.52	1.80
(d)	BERT + ResNext-101	0.1532	0.2248	0.1194	$0.1576^{*,2}$	$0.2288^{\dagger,1}$	$(0.1126)^{\dagger,2}$	6.7	1.70
(e)	W2V + ResNet152	0.1464	(0.2033)	(0.0986)	$0.1507^{*,1}$	$(0.2062)^{*,1}$	$(0.1043)^{\dagger,2}$	6.64	1.71
(f)	BERT + ResNet152	0.1501	(0.2141)	(0.103)	$0.1464^{\dagger,2}$	$(0.208)^{*,2}$	$(0.099)^{\dagger,2}$	6.68	1.71
(g)	Early fusion of W2V + BERT + ResNext-101 + ResNet-152	0.1614	0.2312	0.122	$0.1544^{\dagger,2}$	$0.2327^{\dagger,1}$	$0.12^{*,1}$	9.1	1.73
(h)	(g) + *Att*	0.1635	**0.2427**	**0.1266**	$0.1594^{\dagger,2}$	$0.2444^{\dagger,2}$	$0.1261^{*,2}$	9.21	1.84
(i)	(g) + *Atv*	0.1637	0.2352	0.1265	$0.1583^{\dagger,2}$	$0.2307^{\dagger,2}$	$0.1265^{*,3}$	9.2	1.84
(j)	Late fusion of (a), (d), (e), (f)	0.1658	0.2414	0.1206	0.1683	0.2499	0.1272	26.7	6.84
(k)	Late fusion of (b), (d)+*Att*, (e)+*Att*,(f)+*Att*	**0.1663**	0.2413	0.1240	0.1658	0.2469	0.1283	27.6	7.01
(l)	Late fusion of (c), (d)+*Atv*, (e)+*Atv*, (f)+*Atv*	0.1655	0.2415	0.1245	**0.1693**	**0.2576**	**0.1288**	27.4	7.00

† Adam optimizer   * RMSprop optimizer   [1] learning rate: 1×10^{-4} [2] learning rate: 5×10^{-5} [3] learning rate: 1×10^{-5}

W2V for short] and BERT) and one or both visual representations (i.e., ResNext-101 and ResNet-152). The *Combination of 6 setups* column presents the results after late fusion of 6 different experimental setups for the same network, using two optimizers, i.e. Adam and RMSprop, and 3 learning rates (1×10^{-4}, 5×10^{-5}, 1×10^{-5}), similarly to [16][24], while the *Best of 6 setups* column presents the results of the best-performing among these setups.

The results reported in Table 1(a)-(f) show that both attention mechanisms improve the performance of the baseline method. Furthermore, using better word embeddings (BERT) consistently improves the performance in comparison to W2V.

In the (g), (h) and (i) configurations of Table 1, the results of the early fusion of the text and visual embeddings are presented. The results indicate that the combination of different visual (ResNext-101, ResNet-152) and textual (W2V, BERT) features leads to improved performance. Moreover, the integration of the aforementioned attention mechanisms further improves performance.

Subsequently, in configurations (j), (k) and (l) of Table 1 the performance of late-fusion combinations of the previously-examined networks is presented. In configuration (j), the late fusion of the baseline network trained with different textual and visual features is presented. When considering the combination of 6 setups, this approach usually outperforms by a small margin the corresponding early fusion model (g). The late fusion of models with text- or visual-based attention performs similarly to, or a bit better when combining the best of 6 setups (columns II) compared to the corresponding early fusion approaches, however at the expense of considerably higher training and inference times.

In Table 2 the recommended single-setup early fusion configurations of the proposed method (shaded rows (h) and (i) of Table 1) are compared with the literature SoA works (included the top-performer of the TRECVID 2018 competition [15]), based on the results reported in the corresponding papers for the same evaluation datasets. We can see that the proposed method's configurations (h), (i), outperform the SoA published results on all three datasets.

Table 2: Comparison with published SoA results (MXinfAP).

Method	AVS16	AVS17	AVS18
VSE++ [9]	0.123	0.154	0.074
Video2vec [12]	0.087	0.150	-
W2VV++ [15]	0.151	0.220	0.121
Dual encoding [8]	0.159	0.208	-
(h) from Table 1	**0.1594**	**0.2444**	0.1261
(i) from Table 1	0.1583	0.2307	**0.1265**

5 CONCLUSIONS

This paper examined the problem of video retrieval using textual queries. We focused on a network that encodes the visual and text modalities into a common space. We extended this network by integrating a self-attention mechanism in each modality. The experimental results confirm the contribution of this extension to the performance of the network. Moreover, the effectiveness of using multiple textual and visual representations was experimentally evaluated, and the early fusion of the different text and visual encodings, together with an attention mechanism, was shown to achieve state of the art results without considerable impact on the time-efficiency of the network's training and inference.

ACKNOWLEDGMENTS

This work was supported by the EU Horizon 2020 research and innovation programme under grant agreement H2020-780656 ReTV.

REFERENCES

[1] George Awad, Asad Butt, Keith Curtis, Yooyoung Lee, Jonathan Fiscus, Afzal Godil, Andrew Delgado, Alan F Smeaton, Yvette Graham, Wessel Kraaij, and Georges Quénot. 2019. TRECVID 2019: An evaluation campaign to benchmark Video Activity Detection, Video Captioning and Matching, and Video Search & retrieval. In *TRECVID 2019 Workshop. Gaithersburg, MD, USA*. NIST, USA.

[2] Maria Barrett, Joachim Bingel, Nora Hollenstein, Marek Rei, and Anders Søgaard. 2018. Sequence classification with human attention. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*. 302–312.

[3] Muhammet Bastan, Xiangxi Shi, Jiuxiang Gu, Zhao Heng, Chen Zhuo, Dennis Sng, and Alex Kot. 2018. NTU ROSE Lab at TRECVID 2018: Ad-hoc Video Search and Video to Text. In *TRECVID 2018 Workshop. Gaithersburg, MD, USA*. NIST, USA.

[4] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1724–1734.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805* (2018).

[6] Jianfeng Dong, Xirong Li, and Cees G. M. Snoek. 2016. Word2VisualVec: Image and video to sentence matching by visual feature prediction. *arXiv preprint arXiv:1604.06838* (2016).

[7] Jianfeng Dong, Xirong Li, and Cees G. M. Snoek. 2018. Predicting visual features from text for image and video caption retrieval. *IEEE Transactions on Multimedia (TMM)* 20, 12 (Dec 2018), 3377–3388.

[8] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. 2019. Dual Encoding for Zero-Example Video Retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 9346–9355.

[9] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2018. VSE++: Improving Visual-Semantic Embeddings with Hard Negatives. In *Proceedings of the British Machine Vision Conference (BMVC)*.

[10] Danny Francis, Phuong Anh Nguyen, Benoit Huet, and Chong-Wah Ngo. 2019. Fusion of multimodal embeddings for ad-hoc video search. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. 1868–1872.

[11] Amirhossein Habibian, Thomas Mensink, and Cees GM Snoek. 2014. Composite Concept Discovery for Zero-Shot Video Event Detection. In *Proceedings of the 2014 International Conference on Multimedia Retrieval* (Glasgow, United Kingdom) *(ICMR '14)*. 17–24.

[12] Amirhossein Habibian, Thomas Mensink, and Cees GM Snoek. 2017. Video2vec Embeddings Recognize Events When Examples Are Scarce. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 10 (Oct 2017), 2089–2103. https://doi.org/10.1109/TPAMI.2016.2627563

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[14] Duy-Dinh Le, Sang Phan, Vinh-Tiep Nguyen, Benjamin Renoust, Tuan A Nguyen, Van-Nam Hoang, Thanh Duc Ngo, Minh-Triet Tran, Yuki Watanabe, Martin Klinkigt, et al. 2016. NII-HITACHI-UIT at TRECVID 2016. In *TRECVID 2016 Workshop. Gaithersburg, MD, USA*.

[15] Xirong Li, Chaoxi Xu, Gang Yang, Zhineng Chen, and Jianfeng Dong. 2019. W2VV++: Fully Deep Learning for Ad-hoc Video Search. In *Proceedings of the 27th ACM International Conference on Multimedia*. ACM, 1786–1794.

[16] Xirong Li, Jinde Ye, Chaoxi Xu, Shanjinwen Yun, Leimin Zhang, Xun Wang, Rui Qian, and Jianfeng Dong. 2019. Renmin University of China and Zhejiang Gongshang University at TRECVID 2019: Learn to Search and Describe Videos. In *TRECVID 2019 Workshop. Gaithersburg, MD, USA*.

[17] Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. 2016. TGIF: A new dataset and benchmark on animated GIF description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4641–4650.

[18] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *Proceedings of the 5th International Conference on Learning Representations*.

[19] Foteini Markatopoulou, Damianos Galanopoulos, Vasileios Mezaris, and Ioannis Patras. 2017. Query and keyframe representations for ad-hoc video search. In *Proceedings of the 2017 ACM International Conference on Multimedia Retrieval (ICMR '17)*. ACM, 407–411.

[20] Tomas Mikolov, G.s Corrado, Kai Chen, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, Workshop Track Proceedings (ICLR '13)*.

[21] Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metze, and Amit K. Roy-Chowdhury. 2018. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval (ICMR '18)*. ACM, 19–27.

[22] Niluthpol Chowdhury Mithun, Sujoy Paul, and Amit K Roy-Chowdhury. 2019. Weakly supervised video moment retrieval from text queries. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 11592–11601.

[23] Alan F. Smeaton, Paul Over, and Wessel Kraaij. 2006. Evaluation campaigns and TRECVid. In *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval* (Santa Barbara, California, USA) *(MIR '06)*. ACM Press, New York, NY, USA, 321–330. https://doi.org/10.1145/1178677.1178722

[24] Cees GM Snoek, Xirong Li, Chaoxi Xu, and C. Dennis Koelma. 2017. University of Amsterdam and Renmin University at TRECVID 2017: Searching Video, Detecting Events and Describing Video. In *TRECVID 2017 Workshop. Gaithersburg, MD, USA*.

[25] Kazuya Ueki, Yu Nakagome1, Koji Hirakawa, Kotaro Kikuchi, Tetsuji Ogawa, and Tetsunori Kobayashi. 2017. Waseda Meisei at TRECVID 2017: Ad-hoc video search. In *TRECVID 2017 Workshop. Gaithersburg, MD, USA*.

[26] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1492–1500.

[27] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5288–5296.