SEA: Sentence Encoder Assembly for Video Retrieval by Textual Queries

Xirong Li, *Member, IEEE*, Fangming Zhou, Chaoxi Xu, Jiaqi Ji, Gang Yang

***Abstract*—Retrieving unlabeled videos by textual queries, known as Ad-hoc Video Search (AVS), is a core theme in multimedia data management and retrieval. The success of AVS counts on cross-modal representation learning that encodes both query sentences and videos into common spaces for semantic similarity computation. Inspired by the initial success of previously few works in combining multiple sentence encoders, this paper takes a step forward by developing a new and general method for effectively exploiting diverse sentence encoders. The novelty of the proposed method, which we term *Sentence Encoder Assembly* (SEA), is two-fold. First, different from prior art that use only a single common space, SEA supports text-video matching in multiple encoder-specific common spaces. Such a property prevents the matching from being dominated by a specific encoder that produces an encoding vector much longer than other encoders. Second, in order to explore complementarities among the individual common spaces, we propose multi-space multi-loss learning. As extensive experiments on four benchmarks (MSR-VTT, TRECVID AVS 2016-2019, TGIF and MSVD) show, SEA surpasses the state-of-the-art. In addition, SEA is extremely ease to implement. All this makes SEA an appealing solution for AVS and promising for continuously advancing the task by harvesting new sentence encoders.**

***Index Terms*—Ad-hoc video search, cross-modal representation learning, sentence encoder assembly, multiple space learning**

I. Introduction

VIDEO is arguably the most engaging type of digital content in our society. Research related to video content understanding and retrieval is essential for multimedia data management and retrieval. On one hand, common users have been well educated by web search giants such as Google and Baidu to express their information need in textual queries. While on the other hand, there is an increasing amount of videos lacking reliable annotations or even completely unlabeled. This paper targets at the challenging problem of *ad-hoc video search* (AVS), which is to search on many unlabeled videos for user queries expressed exclusively by a phrase or a natural-language sentence and provided on the fly. The complexity of queries varies, ranging from specific objects, *e.g.,* "a sewing machine", to multi-object events occurred in specific scenes, *e.g.,* "one or more people eating food at a table indoors", see Fig. 1. A cross-modal similarity model that effectively computes the semantic relevance of the unlabeled videos with respect to a given query is crucial. Also, due to the ad-hoc nature of the query, the model has to be generalizable to handle novel queries unseen when the model is built.

For building such a model, both queries and videos have to be encoded into real-valued vectors via cross-modal representation learning. Earlier efforts struggle to detect semantic concepts from the two modalities and use the detected concepts as an intermediate representation [6]–[11]. Now, it is becoming increasingly evident that learning cross-modal representations in an end-to-end and *concept-free* manner is preferred, as manifested via major benchmarks for the AVS task including TRECVID [12]–[14] and MSR-VTT [15]–[17].

We concentrate on end-to-end *query representation learning*, an essential component for AVS. Typically, the component is composed of a sentence encoder that vectorizes a textual query into a constant-sized vector and a feed-forward neural network that projects the vector into a common latent space [12], [15], [18], [19]. Varied types of sentence encoders have been investigated in the growing literature. The vanilla Bag-of-Words (BoW) model is employed by [12], [14], [20], with word2vec (w2v) in [12], [21], GRU / bi-GRU in [12], [14], [15], [22], NetVLAD in [13], [16], and BERT in [17], [23]. While the existing works mainly count on a single sentence encoder, the importance of exploiting multiple sentence encoders for addressing ad-hoc queries has been recognized by few recent works [12]–[14]. The W2VV++ model proposed by Li *et al.* [12] processes a given query by three encoders, *i.e.,* BoW, w2v and GRU, in parallel, and then merges the three encoding results by vector concatenation. Dong *et al.* [14] and their follow-up [13], [24] develop multi-level encoding, where specific sentence encoders are selectively used at distinct levels. Again, vector concatenation is used to combine encodings from the multiple levels. Despite their state-of-the-art performance, we argue that such a concatenation-based method is suboptimal due to the following two reasons. First, the overall encoding could be easily dominated by a specific encoder that produces an encoding vector much longer than the others. For instance, the size of a BoW vector goes up to ten thousand with ease, while encodings of word2vec, GRU or BERT are more compact, with a typical size of a few hundreds. Second, varied encodings by distinct encoders are fed as a whole into the subsequent feed-forward network, meaning the exploration of complementarities among the encoders is limited to a single common space.

In this paper we advance AVS with the following contributions:

Manuscript received August 14, 2020; revised November 05, 2020; accepted November 22, 2020.

This work was supported in part by NSFC (No. 61672523), Beijing Natural Science Foundation (No. 4202033), the Fundamental Research Funds for the Central Universities and the Research Funds of Renmin University of China (No. 18XNLG19). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Raouf Hamzaoui.

The authors are with the Key Lab of Data Engineering and Knowledge Engineering, Renmin University of China, and the AI & Media Computing Lab, School of Information, Renmin University of China, Beijing 100872, China (e-mail: xirong@ruc.edu.cn)

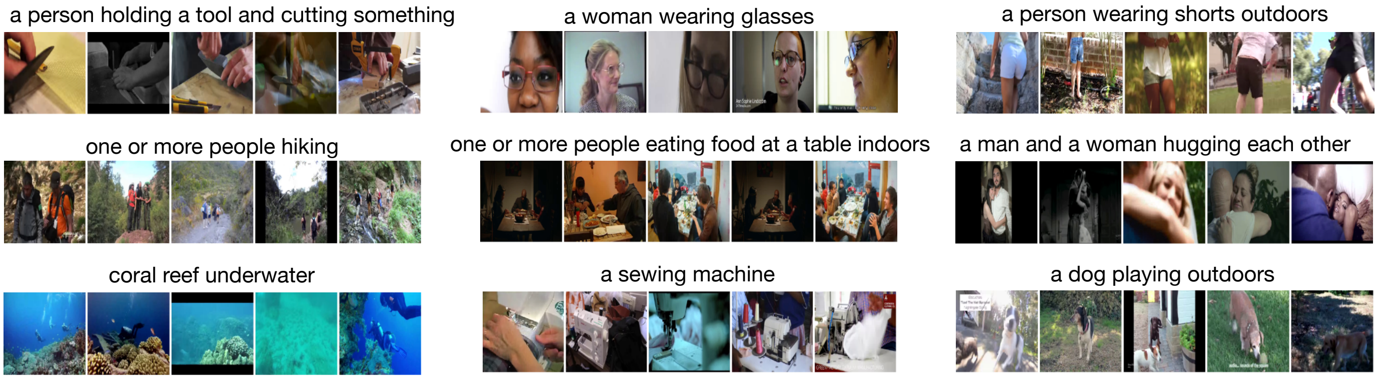

a person holding a tool and cutting something
a woman wearing glasses
a person wearing shorts outdoors
one or more people hiking
one or more people eating food at a table indoors
a man and a woman hugging each other
coral reef underwater
a sewing machine
a dog playing outdoors

Fig. 1. **Top-5 videos per query sentence retrieved from V3C1 [1], a large collection of one million *unlabeled* web video clips, by the proposed *SEA* model**. Queries are from the TRECVID Ad-hoc Video Search benchmarks [2]–[5].

- We propose *Sentence Encoder Assembly* (SEA), a new and general method for effectively exploiting varied sentence encoders. *SEA* bypasses the issues of vector concatenation by learning common spaces per encoder.
- To derive a cross-modal similarity from multiple common spaces, we propose multi-space multi-loss learning, an effective mechanism to explore complementarities among the individual common spaces.
- Our solution surpasses the state-of-the-art on four benchmarks, *i.e.,* MSR-VTT, TRECVID AVS 2016–2019, TGIF and MSVD. Moreover, our solution is easy to implement. With its generality, effectiveness and simplicity, *SEA* has opened up a promising avenue for harnessing novel sentence encoders for continuous performance improvement of AVS. Code and data are available at https://github.com/li-xirong/sea.

II. Related Work

Earlier methods for AVS follow a concept-based approach [8], [9], [25]–[29], with both queries and videos represented in a pre-defined concept space. An intrinsic drawback of the concept-based approach is that concepts in use have to be specified in advance, typically according to their occurrence in training data. Such a hand-crafted common space is suboptimal for cross-modal similarity computation [12]. In order to overcome the drawback, end-to-end learning of concept-free and cross-modal representations has been the mainstream [14]–[16], [30], [31]. As we target at query representation learning, in what follows we discuss recent progress in this direction.

The classical Bag-of-Words (BoW) representation is commonly used for its simplicity. In [20], for instance, a query is first encoded as a BoW vector, and then projected into a latent space through a fully connected layer. However, the BoW encoder has two intrinsic issues. First, it cannot handle semantic relatedness between words. In a BoW feature space, the distance of "a beagle is running" to "a dog is running" is the same as to "a person is running", even though the former pair is visually and semantically more close. Second, it fully ignores word order. To resolve the first issue, Dong *et al.* [32] employ a pre-trained word2vec model to encode each word in a given query into a dense vector and consequently obtain the query vector by mean pooling over the word-level vectors. Later in Liu *et al.* [16], NetVLAD [33] is adopted to exploit second-order statistics of the word-level vectors. To overcome the limit of BoW in sequential modeling, varied forms of sequence-aware deep neural networks are investigated. For instance, GRU is used in [15], bi-LSTM in [34], relational GCN in [30], and more recently BERT in [17], [23].

While more advanced sentence encoders are being actively exploited for AVS, it appears to us that no specific encoder is ready to rule them all. We attribute this to the variety and complexity of AVS queries, which can be short phrases or detailed descriptions of multiple-object actions in specific scenes. For the former case, a BoW encoder will suffice, while the latter case requires a complicated encoder to effectively capture fine-grained information. In the context of image/video caption retrieval, Dong *et al.* [32] make an initial endeavor to combine multiple encoders including BoW, w2v and GRU for query representation. In particular, they concatenate the output of the individual encoders into a lengthy vector. Based on [32], Li *et al.* [12] develop W2VV++, the winning entry for the TRECVID AVS 2018 evaluation [35]. Contemporarily, Dong *et al.* [14] propose the Dual Encoding network, wherein three encoders, *i.e.,* BoW, bi-GRU and 1-d CNN, are employed to build a multi-level query representation. Follow-ups of [14], *e.g.,* [13], [24], [31] also leverage multiple encoders, and again merge the output of the encoders in advance to cross-modal representation learning. By contrast, our multi-space learning mechanism makes our model more flexible to harness the complementarities between distinct sentence encoders. Consequently, even with common 2D-CNN features as video representation, our proposed model compares favorably against the state-of-the-art.

Note that at a high level, the idea of sentence encoder assembly is similar to the conventional ensemble methods [36]. However, ensemble learning is a very general idea, typically studied in the context of a classification task. Therefore, a gap naturally exists between the idea itself and putting it to work on AVS. This paper is an initial attempt to bridge the gap.

III. Proposed Method

A. Problem Formalization

We formalize an ad-hoc video search process as follows. We denote a specific video clip as v and a large collection

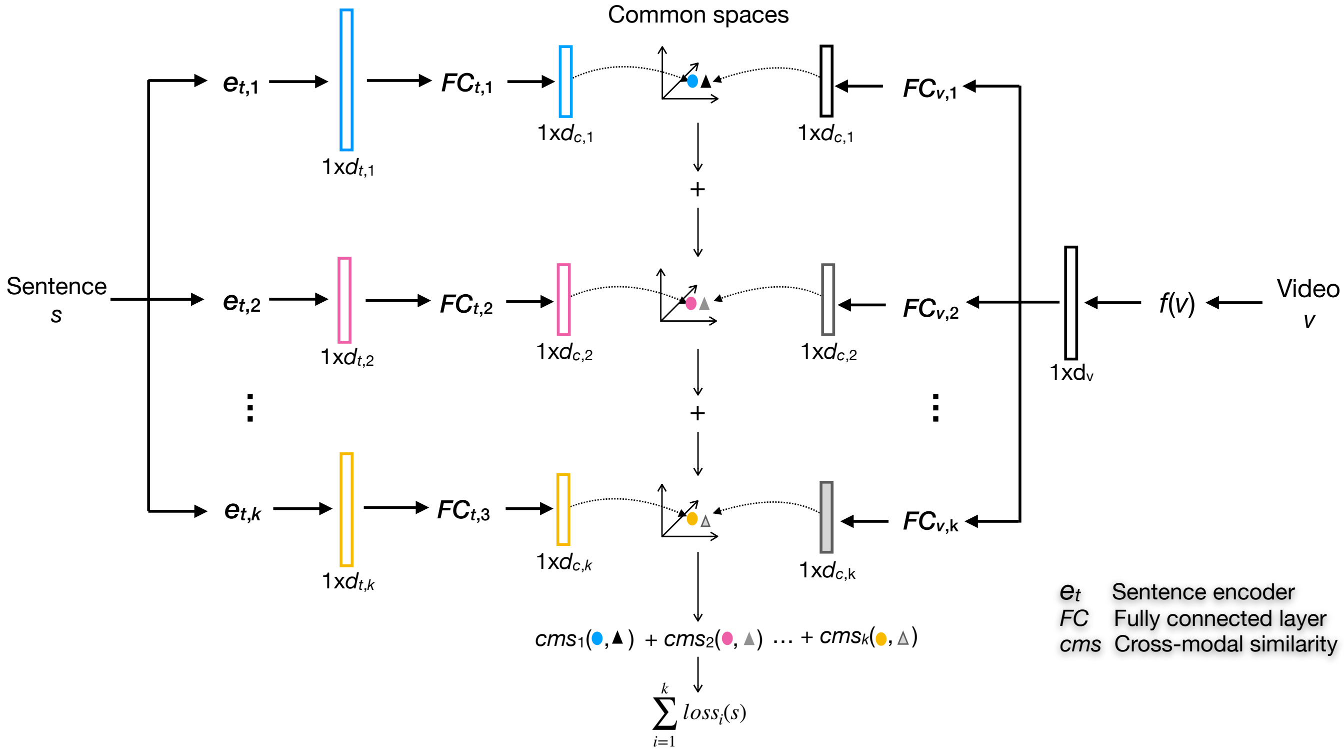


Fig. 2. **Proposed Sentence Encoder Assembly (*SEA*) method for exploiting multiple sentence encoders $\{e_{t,1},\ldots,e_{t,k}\}$ for computing cross-modal similarities between a given query sentence s and a specific unlabeled video v.** Instead of concatenating the output of the individual sentence encoders as in previous works [12]–[14], our *SEA* model simultaneously learns k common spaces for the k encoders. Rather than minimizing a single loss computed based on the combined similarity $\sum_{i=1}^{k} cms_i(s,v)$, *SEA* is trained to minimize a combine loss $\sum_{i=1}^{k} loss_i(s)$. Such a multi-space multi-loss learning mechanism is novel and crucial for AVS, meanwhile easy to implement.

of n *unlabeled* video clips as $\mathcal{V}=\{v_1,\ldots,v_n\}$. For an ad-hoc query in the form of a sentence s, let $cms(s,v)$ be a **c**ross-**m**odal **s**imilarity function that measures the semantic relevance between the query and a specific video. Accordingly, the search process boils down to sorting $\mathcal{V}$ in descending order in terms of $cms(s,v)$ and returning the top ranked items for the given query. The computation of $cms(s,v)$ requires proper embeddings of both s and v into a common cross-modal space. While visual CNNs are prerequisites for video embedding, sentence encoders are required for query embedding. Let e_t be a specific sentence encoder, which encodes the given query into a d_t-dimensional real-valued vector, *i.e.,* $e_t(s)\in R^{d_t}$. Having k distinct sentence encoders $\{e_{t,1},\ldots,e_{t_k}\}$ shall give us k vectors of varied dimensions $\{d_{t,1},\ldots,d_{t,k}\}$. We aim for a model that effectively exploit the multiple sentence encoders for computing $cms(s,v)$.

Next, we describe in brief sentence encoders investigated in this work in Section III-B, followed by the proposed sentence encoder assembly (SEA) model in Section III-C.

B. Sentence Encoders in Use

We consider five present-day sentence encoders, *i.e.,* Bag-of-Words (BoW), word2vec (w2v), GRU, bi-GRU and BERT. Among them, the first two are unigram, while the others are sequential models. Their main properties are summarized in Table I.

1) BoW. As a classical text encoder, BoW simply quantizes a given sentence s of l words with respect to a pre-specified vocabulary of m words. Let $c(s,j)$ be a function that counts the occurrence of the j-th word in the sentence. Accordingly, we have the BoW encoding $e_{BoW}(s)$ as

$$e_{BoW}(s) := (c(s,1),\ldots,c(s,m)). \tag{1}$$

Note that an AVS query is relatively short, often containing less than 10 words. Meanwhile, the vocabulary size is much larger, with a typical order of 10^4. As a consequence, $e_{BoW}(s)$ is a long and sparse vector.

2) w2v. The w2v model [37] learns to produce word-level dense and semantic vectors by training a two-layer neural network on a large text corpus, with the goal to reconstruct

TABLE I

FIVE SENTENCE ENCODERS USED IN THIS PAPER. THE SPECIFIC VALUE OF THE VOCABULARY SIZE m IS DATASET-DEPENDENT, WHICH IS 7,676 FOR MSR-VTT, 3,981 FOR TGIF AND 2,917 FOR MSVD. THE NOTATION $m+$ MEANS THE VOCABULARY OF GRU AND BI-GRU IS SLIGHTLY BIGGER THAN m DUE TO THE INCLUSION OF STOPWORDS AND SPECIAL TOKENS FOR SEQUENTIAL MODELING.

Encoder	**Vocabulary**	**Dim.** d_t	**Training**	**Prior work**
BoW	m	m	Not trainable	[12], [14], [20], [32]
w2v	1.7 millions	500	pre-trained[1] and fixed	[12], [13], [21], [32]
GRU	$m+$	1,024	trained from scratch	[12], [15], [32]
bi-GRU	$m+$	2,048	trained from scratch	[13], [14]
BERT	30,000	768	pre-trained[2] and fixed	[17], [23]

linguistic contexts of words in the training text. As computing the reconstruction loss requires no extra manual annotation, w2v encodes millions of words with ease. We adopt a 500-dimensional w2v model[1] from [32]. We also tried alternatives such as GloVe [38], and found it less effective in our preliminary experiments. Let $w2v(s[i])$ be a lookup function that returns the embedding vector for the i-th word of s, we obtain w2v based sentence encoding by mean pooling, *i.e.,*

$$e_{w2v}(s) := \frac{1}{l}\sum_{i=1}^{l} w2v(s[i]). \tag{2}$$

3) GRU. The Gated Recurrent Unit (GRU) network [39] models the sequential information within a sentence by iteratively generating a sequence of recurrent hidden state vectors $\{\vec{h}_1, \ldots, \vec{h}_l\}$. In particular, the hidden state vector at time-step i, $\vec{h}_i$, is jointly determined by the word embedding of the current word $s[i]$ and $\vec{h}_{i-1}$, the hidden state vector at the previous time-step. Similar to LSTM [40], the GRU network effectively prevents the vanishing gradient problem by introducing a gating mechanism to modulate the flow of information inside the unit. Meanwhile, as GRU has no separate memory cell, it has a simplified architecture and thus with less parameters to be trained. Following [12], we obtain the GRU-based sentence encoding by mean pooling over the hidden vector sequence, *i.e.,*

$$e_{gru}(s) := \frac{1}{l}\sum_{i=1}^{l} \vec{h}_i. \tag{3}$$

4) bi-GRU. The bi-directional GRU (bi-GRU) network extends the forward GRU by including a backward GRU that encodes the sequence in a reverse order. Given $\{\overleftarrow{h}_1, \ldots, \overleftarrow{h}_l\}$ as hidden state vectors of the backward GRU, our bi-GRU based sentence encoding is obtained by

$$e_{bigru}(s) := \frac{1}{l}\sum_{i=1}^{l} \vec{h}_i \oplus \overleftarrow{h}_i, \tag{4}$$

where $\oplus$ denotes vector concatenation. Note that given forward and backward hidden vectors of the same size, e_{bigru} provides a richer representation than e_{gru} at the cost of doubled parameters. Hence, we shall use either e_{gru} or e_{bigru}, but not both.

5) BERT. The BERT model, built by stacking a number of L bi-directional Transformer blocks [41], generates word embeddings for a given sentence by progressively passing encodings through the multiple blocks. A Transformer block consists of a self-attention network and a feed-forward network [42]. The self-attention network accepts encodings of individual tokens from the previous Transformer block, weighs their importance to each other by a self-attention mechanism, and accordingly generates new encodings. These encodings are then fed in parallel into the feed-forward network to produce the output encodings of this block. In this work, we adopt the base version of BERT containing $L = 12$ blocks, which has been pre-trained on English Wikipedia and book corpora for masked language modeling and next sentence prediction[2]. We obtain the BERT-based sentence encoding by mean pooling as

$$e_{bert}(s) = \frac{1}{l}\sum_{i=1}^{l} \text{token-emb}(s[i], L-1), \tag{5}$$

where token-emb$(s[i], L-1)$ denotes the embedding of the i-th word produced by the second-last block. Note that we tried max pooling or using the embedding of the first / last token, and found these alternatives less effective than mean pooling.

With the sentence encoders introduced, we proceed to describe how to effectively combine them in an end-to-end framework.

C. Sentence Encoder Assembly

We propose to combine k distinct sentence encoders $\{e_{t,i} | i = 1, \ldots, k\}$ in a generic multi-space multi-loss learning framework.

Multiple common spaces. Our framework consists of k cross-modal matching subnetworks, each corresponding to a specific sentence encoder and learning its own common space. Each subnetwork, indexed by i, consists of two fully connected (FC) layers, one on the text side to transform $e_{t,i}(s)$ into a $d_{c,i}$-dimensional vector, and the other on the video side that transforms the video feature vector $f(v)$ into another $d_{c,i}$-dimensional vector. Consequently, the sentence-video semantic relevance, denoted as $cms_i(s, v)$, is computed as the cosine similarity between the two embedding:

$$cms_i(s, v) := \text{cosine-sim}(\underbrace{FC_{t,i}(e_{t,i}(s))}_{\text{text embedding}}, \underbrace{FC_{v,i}(f(v))}_{\text{video embedding}}), \tag{6}$$

where $FC_{t,i}$ and $FC_{v,i}$ indicate the two FC layers, each followed by a *tanh* function to increase their learning capacity. We choose the cosine similarity as it is a widely used similarity metric for cross-modal matching [14], [16], [18], [43], [44]. We also tried a Euclidean distance based similarity, which is however less effective[3].

By simply averaging the similarities computed in the individual common spaces, we have the overall cross-modal similarity as

$$cms(s, q) := \frac{1}{k}\sum_{i=1}^{k} cms_i(s, v). \tag{7}$$

Note that we do not go for more complicated alternatives, *e.g.,* weighing the individual similarities by self-attention mechanisms. Rather, we opt for this simple combination strategy, not only for preventing the risk of over-fitting. Such a strategy also encourages the individual common spaces to be good enough to be combined, as they are set to be equally important.

Multi-loss learning. We develop our loss function based on the improved triplet ranking loss (ITRL) by Faghri *et al.* [43].

[1] https://github.com/danieljf24/w2vv Note that while [32] performs image-to-text matching experiments on Flickr30k, its w2v model was trained on English tags of 30 million Flickr images, using the skip-gram algorithm.

[2] https://github.com/google-research/bert

[3] For *SEA* ({BoW,w2v}) trained with the Euclidean distance based similarity, its infAP scores on TV16/17/18/19 are 12.8/18.9/11.8/10.4, clearly lower than the cosine similarity counterpart (15.7/23.4/12.8/16.6).

While originally proposed for image-text matching, ITRL is now found to be effective for text-video matching [12]–[16]. Unlike the classical triplet ranking loss that selects negative training examples by random, ITRL considers the negative that violates the ranking constraint the most (within a mini-batch) and thus deemed to be the most informative for improving the model being trained. Given a training sentence s with v^+ as a video relevant w.r.t s and v^- as irrelevant, we express ITRL as

$$\left\{\begin{array}{ll} v^{-*} & = \mathrm{argmax}_{v^- \in batch}(cms(s,v^-) - cms(s,v^+)) \\ ITRL(s) & = \max(0, \alpha + cms(s,v^{-*}) - cms(s,v^+)), \end{array}\right. \tag{8}$$

where α is a positive hyper-parameter concerning the margin.

We argue that such a single loss is suboptimal for multi-space learning. Given a specific mini-batch, hard negative examples selected in terms of the combined similarity are not necessarily the most effective for learning the individual common spaces. Therefore, we choose to compute $ITRL_i(s)$ per space, and accordingly learn to minimize their combined loss, *i.e.,*

$$\sum_{i=1}^{k} ITRL_i(s). \tag{9}$$

In a similar spirit to similarity combination, we again treat all the sub losses equally. As exemplified in Fig. 3, the combined loss lets the model be exposed to more diverse hard negatives. We empirically find that compared to the single loss, the combined loss provides around 30% extra hard negatives per training epoch.

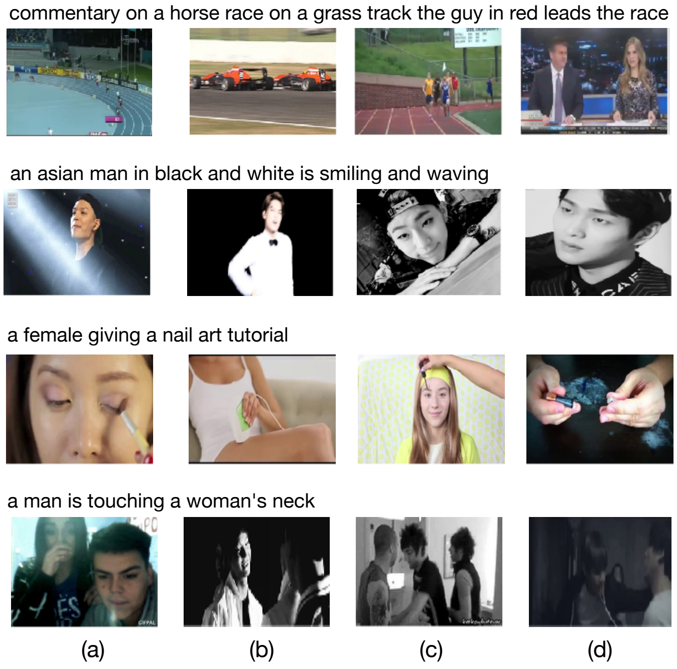

commentary on a horse race on a grass track the guy in red leads the race
an asian man in black and white is smiling and waving
a female giving a nail art tutorial
a man is touching a woman's neck
(a)
(b)
(c)
(d)

Fig. 3. **Examples of hard negative videos automatically selected for specific sentences during training**. The first column (a) is selection based on the combined similarity in a single common space. The other columns indicate selections made based on individual similarities w.r.t (b) e_{BoW}, (c) e_{w2v}, and (d) e_{gru} within the proposed multi-space and multi-loss framework. Using the combined loss allows the model to be exposed to more diverse hard negatives in a given batch.

TABLE II
DATASETS USED IN OUR EVALUATION. FOR ALL EXPERIMENTS WE TRAIN MODELS ON THE SPECIFIED TRAINING SET AND USE THE CORRESPONDING VALIDATION SET FOR MODEL SELECTION.

Data split	**Data sources**	**Video clips**	**Frames**	**Queries**
MSR-VTT experiments:				
train set	MSR-VTT [45]	6,513	197,648	–
val. set		497	15,347	9,940
test-full		2,990	92,467	59,800
test-1k [34]		1,000	30,932	1,000
TRECVID experiments:				
train set	MSR-VTT [45]	10,000	305,462	–
	TGIF [46]	100,855	1,045,268	–
val. set	TV16-VTT-dev [2]	200	5,941	200
test set for TV16/17/18	IACC.3 [2]	335,944	3,845,221	90
test set for TV19	V3C1 [1]	1,082,649	7,839,450	30
TGIF experiments:				
train set	TGIF [46]	78,799	818,140	–
val. set		10,705	110,252	10,828
test set		11,351	116,876	34,074
MSVD experiments:				
train set	MSVD [47]	1,200	23,313	–
val. set		100	2,415	4,291
test set		670	15,429	27,767

To sum up, the multi-space strategy provides a more flexible mechanism to exploit complementarities among the distinct sentence encoders. Meanwhile, given a specific mini-batch during training, the multi-loss strategy allows each common space to select its own hard negative example. More flexibility in encoder ensemble and more effectiveness for training together contributes to the superior performance of the proposed SEA method against the state-of-the-art.

IV. EVALUATION

We first conduct experiments on two major benchmarks, MSR-VTT [45] and TRECVID AVS [2]. While originally developed for video captioning, MSR-VTT has been adopted by recent works for text-based video retrieval [14]–[18], [30], [34]. TRECVID AVS is a leading benchmark for ad-hoc video search at large-scale since 2016 [2]–[5]. The two benchmarks have their own characteristics. As shown in Table II, while MSR-VTT has a relatively small amount of 2,990 test videos, it has over 59k query sentences. As for TRECVID, it has a much larger number of test videos, over 335k in the 2016 / 2017 / 2018 editions and over one million in the 2019 edition. Hence, a joint evaluation on the two benchmarks provides a comprehensive assessment of the state-of-the-art. In addition, we report performance on TGIF [46] and MSVD [47].

A. Experimental Setup

We first describe experimental setups unique to MSR-VTT and TRECVID, and then introduce common implementations.

1) Setup for MSR-VTT: We follow the official data split, which divides MSR-VTT into three disjoint subsets used for training, validation and test, respectively. Note that in [34] and its follow-ups [16]–[18], a smaller test set of 1,000 videos randomly sampled from the full test set is used, which we refer to as *test-1k*.

Performance metrics. Following the previous works, we report $R@k$, $k = 1, 5, 10$, the percentage of test queries that have at least one relevant video covered in the top k returned items, and Median rank (*Med r*), the median rank of the first relevant video in the search results. Mean Average Precision (*mAP*) is also reported to assess the overall ranking quality.

2) Setup for TRECVID: We evaluate on the TRECVID AVS testbed from the last four years. The test video collection for TV16 / TV17 / TV18 is IACC.3 [2], containing 335,944 web video clips. The test collection for TV19 is V3C1 [48], which contains 1,082,649 web video clips, with even more diverse content, no predominant characteristics and low self-similarity [1]. As no training data is provided by the organizers, we adopt the setup of the winning entry of TV18 [35], using MSR-VTT and TGIF [46] for training and the development set of the TV16 video-to-text matching task [2] for validation.

Performance metric. The official metric, *i.e.,* inferred average precision (infAP) [49], is used.

3) Common Implementations: We use public feature data[4], where each video is represented by a 4,096-d feature vector, obtained by using two pre-trained CNNs, *i.e.,* ResNet-152 and ResNeXt-101, to extract 2,048-d features from video frames. Frame-level features are concatenated and aggregated to video-level features by mean pooling. We refer to [12] for details.

Training. The margin parameter α in the loss is set to 0.2 according to [43]. The dimensionality of all the common spaces $d_{c,i}$ is set to 2,048, which achieves a good balance between model performance and model complexity. In fact, our model is highly robust to the choice of the common space dimensionality, see the Appendix. We perform SGD based training, with a mini-batch size of 128 and RMSProp as the optimizer. The learning rate is initially set to 10^{-4}, decayed by a factor of 0.99 per epoch. Following [50], we half the learning rate if the validation performance does not increase in three consecutive epochs. Early stop occurs when no validation performance increase is achieved in ten consecutive epochs. For each model with specific configurations of sentence encoders, we repeat training three times and pick the version that maximizes the validation performance. All experiments were done with PyTorch (1.2.0) [51] on an Nvidia GEFORCE GTX 1080Ti GPU.

B. Experiment 1. Which Sentence Encoders to Use?

We compare with the state-of-the-art W2VV++ [12], which combines multiple sentence encoders by concatenating their output into a long feature vector and then embeds the concatenated vector into a common space by an FC layer. While originally developed for automated search, W2VV++ has been used with success by Kratochvíl *et al.* [52] and Lokoć *et al.* [23] in the Video Browser Showdown, a leading benchmark for interactive video retrieval [53], [54]. For a fair comparison, we use author-provided source code[5] with the same setup as described in Section IV-A3.

The performance of W2VV++ and the proposed *SEA* model is presented in Table III. For all configurations of sentence encoders, *SEA* consistently outperforms its W2VV++ counterpart. Specifically, on MSR-VTT our model obtains a relative improvement ranging from 3.6% to 7.3% in terms of mAP. While on TRECVID, the relative improvement w.r.t the overall performance ranges from 7.3% to 25.7%. The advantage becomes even more clear when four sentence encoders are combined, see the last two rows in Table III. These results justify the effectiveness of the multi-space mechanism.

As more sentence encoders are included, we observe different phenomenons on the two benchmarks. For MSR-VTT, adding sequential encoders is helpful. Compared to *SEA* ({BoW, w2v}), *SEA* ({BoW, w2v, GRU}) improves mAP from 21.3 to 22.1, while substituting BERT for GRU obtains higher mAP of 23.0. The peak performance, mAP of 23.3, is reached by *SEA* ({BoW, w2v, GRU, BERT}) and *SEA* ({BoW, w2v, bi-GRU, BERT}). By contrast, the inclusion of GRU and BERT has a negative impact on TRECVID. Compared to *SEA* ({BoW, w2v}) which has the best overall infAP of 17.1, adding GRU results in an overall infAP of 16.8, while adding BERT results in a lower value of 16.3. By analyzing the query sentences of the two benchmarks, we find that an MSR-VTT sentence tend to be longer, containing 9.3 words on average, while the corresponding number of TRECVID is 7.1. We attribute this difference to the fact that MSR-VTT was originally meant for video captioning, so its sentences are more detailed. This is furthered confirmed by part-of-speech statistics, where we find that an MSR-VTT query has 3.3 nouns, 1.8 verbs and 0.6 adjective on average, while a TRECVID query has 2.7 nouns, 1.0 verb and 0.4 adjective. TRECVID queries are more keyword-oriented, *e.g.,* “a newspaper”, “people shopping”, and “a blond female indoors”. Hence, for answering keyword-oriented queries, *SEA* ({BoW, w2v}) is most suited, while a full setup, *e.g., SEA* ({BoW, w2v, bi-GRU, BERT}), is preferred for addressing description-oriented queries.

We further analyze the complementarity between the distinct sentence encoders by inspecting how they behave when used individually. To that end, we train a cross-modal matching network per encoder on MSR-VTT. To obtain an intuitive understanding of what each network has learned as its common space, we perform sentence-to-sentence retrieval, using all the 200k captions in MSR-VTT as a sentence pool. As w2v and BERT are pre-trained on large-scale corpora with a large vocabulary, they better handle subjects of low occurrence (‘beagle’) or zero occurrence (‘rottweiler’) in the training data, see Fig. 4. Interestingly, for the query ‘a is running on lawn’ where we have intentionally remove the subject, BoW returns sentences describing person running, while some of the top-ranked sentences by BERT are still related to dogs. Moreover, we perform a per-query comparison between the matching networks for video retrieval. Among all the 59,800 test queries, the network with BoW is better than the others for 15.2% of the test queries, while the numbers corresponding to w2v, GRU, bi-GRU and BERT are 14.5%, 12.9%, 14.0% and 19.6%, respectively. The results clearly show the complementarity between the encoders.

[4] https://github.com/li-xirong/avs

[5] https://github.com/li-xirong/w2vvpp

TABLE III

JOINT EVALUATION OF SENTENCE ENCODERS AND THEIR ASSEMBLY MODELS, *i.e.*, W2VV++ [12] AND THE PROPOSED *SEA*, ON MSR-VTT AND TRECVID. NUMBERS ARE SHOWN IN PERCENTAGES, WITH BEST SCORES SHOWN IN **BOLD** FONT. FOR A GIVEN SETUP OF SENTENCE ENCODERS, RELATIVE IMPROVEMENT OF *SEA* OVER ITS W2VV++ COUNTERPART IS GIVEN IN PARENTHESES. *SEA* IS CONSISTENTLY BETTER.

Sentence encoders	**Model**	**MSR-VTT** (the full test set)					**TRECVID** (metric: infAP)				
		R@1	*R@5*	*R@10*	*Med r*	*mAP*	*TV16*	*TV17*	*TV18*	*TV19*	*MEAN*
{BoW, w2v}	W2VV++	10.9	29.1	39.9	19	20.2	14.4	21.8	11.1	14.3	15.4
	SEA	11.6	30.6	41.6	17	21.3 (↑5.4%)	15.7	**23.4**	**12.8**	16.6	**17.1** (↑11.2%)
{BoW, w2v, GRU}	W2VV++	11.1	29.6	40.5	18	20.6	16.2	22.3	10.1	13.9	15.6
	SEA	12.2	31.9	43.1	15	22.1 (↑7.3%)	15.0	**23.4**	12.2	16.6	16.8 (↑7.5%)
{BoW, w2v, bi-GRU}	W2VV++	11.3	29.9	40.6	18	20.8	16.1	21.7	10.4	13.5	15.4
	SEA	12.4	32.1	43.3	15	22.3 (↑7.2%)	**16.4**	22.8	12.5	**16.7**	**17.1** (↑10.9%)
{BoW, w2v, BERT}	W2VV++	12.3	31.8	43.0	15	22.2	15.1	22.5	10.2	12.8	15.2
	SEA	12.8	33.1	44.6	**14**	23.0 (↑3.6%)	15.3	22.8	12.1	14.8	16.3 (↑7.3%)
{BoW, w2v, GRU, BERT}	W2VV++	12.1	31.7	42.7	16	22.0	14.3	19.3	9.3	10.1	13.3
	SEA	13.0	**33.6**	44.9	**14**	**23.3** (↑5.9%)	16.0	23.1	12.1	15.4	16.7 (↑25.7%)
{BoW, w2v, bi-GRU, BERT}	W2VV++	12.0	31.3	42.3	16	21.8	15.8	20.6	9.0	10.5	14.0
	SEA	**13.1**	33.4	**45.0**	**14**	**23.3** (↑6.9%)	15.9	22.9	11.7	15.5	16.5 (↑18.1%)

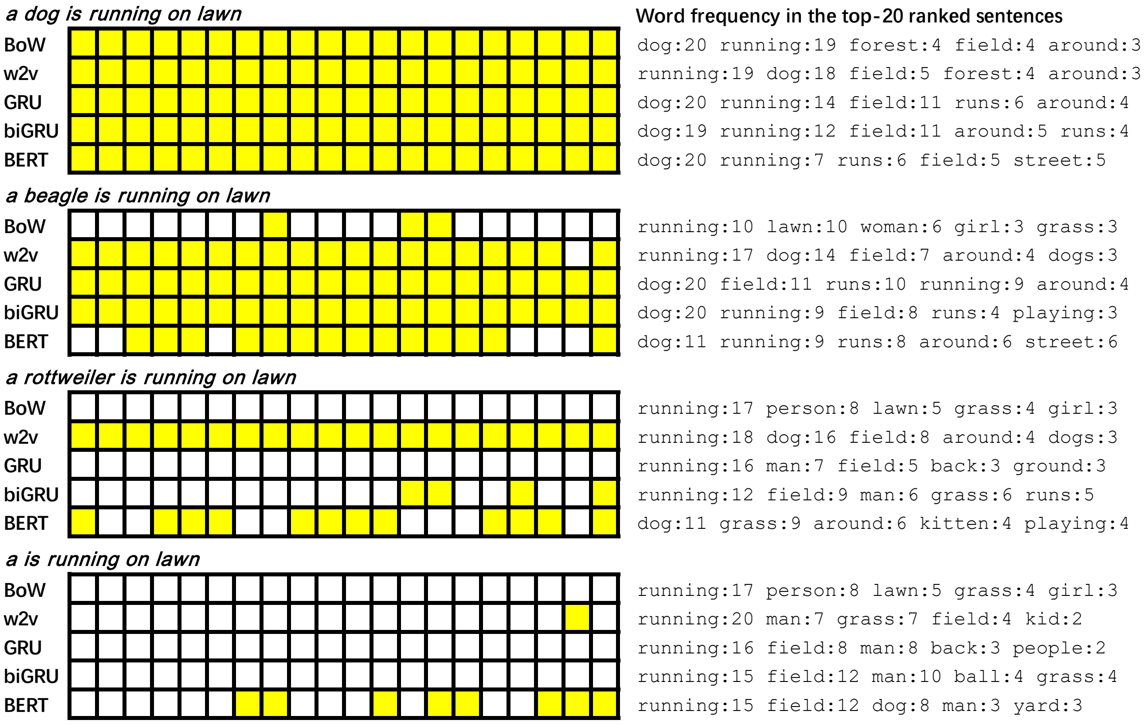


Fig. 4. **Visualization of sentence-to-sentence retrieval results**. Given a query sentence, *e.g.*, "a dog is running on lawn", we retrieve top-20 sentences from MSR-VTT (which has 200k sentences in total), using common spaces learned by cross-modal matching networks with respect to specific sentence encoders. A yellow grid indicates sentences related to dogs. For the last query, we intentionally remove the subject. Encoders pre-trained on large-scale corpora, *i.e.*, w2v and BERT, better handle subjects of low occurrence ('beagle') or zero occurrence ('rottweiler') in the training data.

TABLE IV

EVALUATING DIFFERENT METHODS FOR FUSING MULTIPLE SENTENCE ENCODERS, *i.e.*, {BoW, w2v, GRU}. THE MOST EFFECTIVE METHOD IS TO TRAIN THE *SEA* MODEL WITH THE COMBINED LOSS.

Fusion method	**TV16**	**TV17**	**TV18**	**TV19**	**MEAN**
W2VV++	**16.2**	22.3	10.1	13.9	15.6
Transformed W2VV++	13.9	20.2	10.2	13.5	14.5
Model averaging	14.9	21.9	11.6	15.4	16.0
SEA single loss	14.7	21.8	11.2	14.7	15.6
SEA combined loss	15.0	**23.4**	**12.2**	**16.6**	**16.8**

C. Experiment 2. Other Alternatives for Encoder Assembly?

As the output size of the individual sentence encoders ranges from 500 (e_{w2v}) up to over 10k (e_{BoW}), one might naturally challenge the deficiency of the concatenation operation used by W2VV++. For a more comprehensive comparison, we further implement two more alternatives:

- *Transformed W2VV++*. We modify W2VV++ by adding an FC layer after each encoder to transform all encodings into 2,048-d vectors in advance to concatenation. This allows the size of the concatenated vector to be invariant with respect to the encodings. The new variant is also end-to-end trained.
- *Model averaging*. The cross-modal subnetworks w.r.t the

individual encoders are trained separately. Cross-modal similarities computed by the subnetworks are equally combined.

Table IV shows the results of these alternatives on TRECVID. The lower performance of Transformed W2VV++ suggests that adjusting the encodings causes loss in the original information produced by the individual encoders. Model averaging outperforms W2VV++, again suggesting the benefit of using multiple common spaces against a single common space. The result that model averaging is less effective than *SEA* verifies the necessity of learning multiple common spaces in a unified framework.

D. *Experiment 3. Combined loss versus Single loss*

We have qualitatively illustrate the benefit of the combined loss against the single loss in Fig. 3. Now we provide more quantitative evidence. As Table IV shows, *SEA* trained with the single loss does not outperform W2VV++. We can also observe similar results from the learning curves in Fig. 5. For both W2VV++ and *SEA*, we use {BoW, w2v, GRU} as their sentence encoders. Note that the number of epochs each model takes is not pre-specified. Due to the early stop strategy, the number of training epochs actually took varies among the models. As shown in Fig. 5, *SEA* with the single loss (the blue curve) quickly converged to a suboptimal state. These results proof the importance of the combined loss for training the multi-space network.

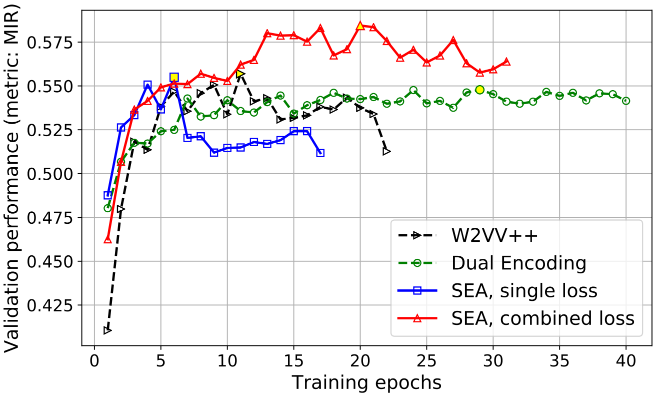


Fig. 5. **Learning curves of distinct models in the TRECVID experiments**. Validation is performed after each epoch. The number of epochs a model takes is not pre-specified. After a model reaches its peak performance, as indicated by yellow markers, early stopping occurs in ten epochs. So the number of training epochs actually took varies among the models. Both W2VV++ and *SEA* use {BoW, w2v, GRU} as their sentence encoders. For training the multi-space network, the combined loss is preferred over the single loss.

E. *Comparison to the State-of-the-Art*

1) On MSR-VTT: We compare with 11 recent models as follows, which have been evaluated on the *test-1k* set [17], [18], [34], the *full* set [30], [31], [55] or both [12], [14], [16], [43], [44]. We highlight their choices of sentence encoders:

- JSFusion [34]: Use bi-LSTM as its sentence encoder.
- VSE++ [43]: Use GRU as its sentence encoder.
- Mithun *et al.* [55]: Use GRU as its sentence encoder.
- Miech *et al.* [18]: Use a 1D-CNN as its sentence encoder.

TABLE V

State-of-the-art on **MSR-VTT** for text-based video retrieval. Best scores from the cited papers are used, where applicable. On both the *test-1k* set and the *full* test set, our proposed *SEA*({BoW, w2v,bi-GRU, BERT}) is the best.

Test set	**Model**	**R@1**	**R@5**	**R@10**	**Medr**	**mAP**
1k [34]	JSFusion [34]	10.2	31.2	43.2	13	n.a.
	VSE++ [43]	15.2	37.7	50.1	10	26.0
	TCE [44]	16.1	38.0	51.5	10	n.a.
	Miech *et al.* [18]	14.9	40.2	52.8	9	n.a.
	UniViLM [17]	15.4	39.5	52.3	9	n.a.
	Dual Encoding [14]	18.8	44.4	57.2	7	31.6
	W2VV++ [12]	18.9	45.3	57.5	8	31.6
	CE [16]	20.9	48.8	62.4	6	n.a.
	SEA	**23.8**	**50.3**	**63.8**	**5**	**36.6**
Full	Mithun *et al.* [55]	7.3	21.7	30.9	34	n.a.
	TCE	7.7	22.5	32.1	30	n.a.
	CF-GNN [31]	8.0	23.2	32.6	31	16.0
	VSE++	8.7	24.3	34.1	28	16.9
	HGR [30]	9.2	26.2	36.5	24	n.a
	CE	10.0	29.0	41.2	16	n.a.
	Dual Encoding	11.1	29.4	40.3	19	20.5
	W2VV++	11.1	29.6	40.5	18	20.6
	SEA	**13.1**	**33.4**	**45.0**	**14**	**23.3**

- Dual Encoding [14]: Hierarchical encoding that combines BoW, bi-GRU and 1D-CNN.
- W2VV++ [12]: Concatenate encodings of BoW, w2v and GRU
- CE [16]: Use NetVLAD as its sentence encoder.
- TCE [44]: Use a latent semantic tree for query representation learning.
- HGR [30]: Encode by hierarchical semantic graph including three levels of events, actions, entities and relationships across levels.
- CF-GNN [31]: Graph neural network based search result reranking, with Dual Encoding as its sentence encoder.
- UniViLM [17]: BERT as its sentence encoder.

Note that all the models were trained on the official training set of MSR-VTT except for [17], [18], where the authors pre-trained their model on 100 million narrated video clips and then fine-tuned on MSR-VTT.

Results. Table V shows the performance of the distinct models on the MSR-VTT full test set and *test-1k*. For the ease of comparison, the performance of the baselines is directly cited from the original papers except for W2VV++[5], VSE++[6] and Dual Encoding[7], which we have re-trained using their public code with the same video feature as used in this work. Among the baselines, CE is the best on *test-1k*, while W2VV++ is the best on the full test set. On both sets, the proposed *SEA* model is the top performer. Notice that the good performance of CE is obtained by representing videos with many features including appearance, scene, motion, face, OCR, speech and audio. Given the simplicity of our video feature, the advantage of the new model is clearly justified.

2) On TRECVID AVS 2016–2019: We compare with the top-3 finalist of the TRECVID AVS evaluation each year,

[6] https://github.com/fartashf/vsepp
[7] https://github.com/danieljf24/dual_encoding

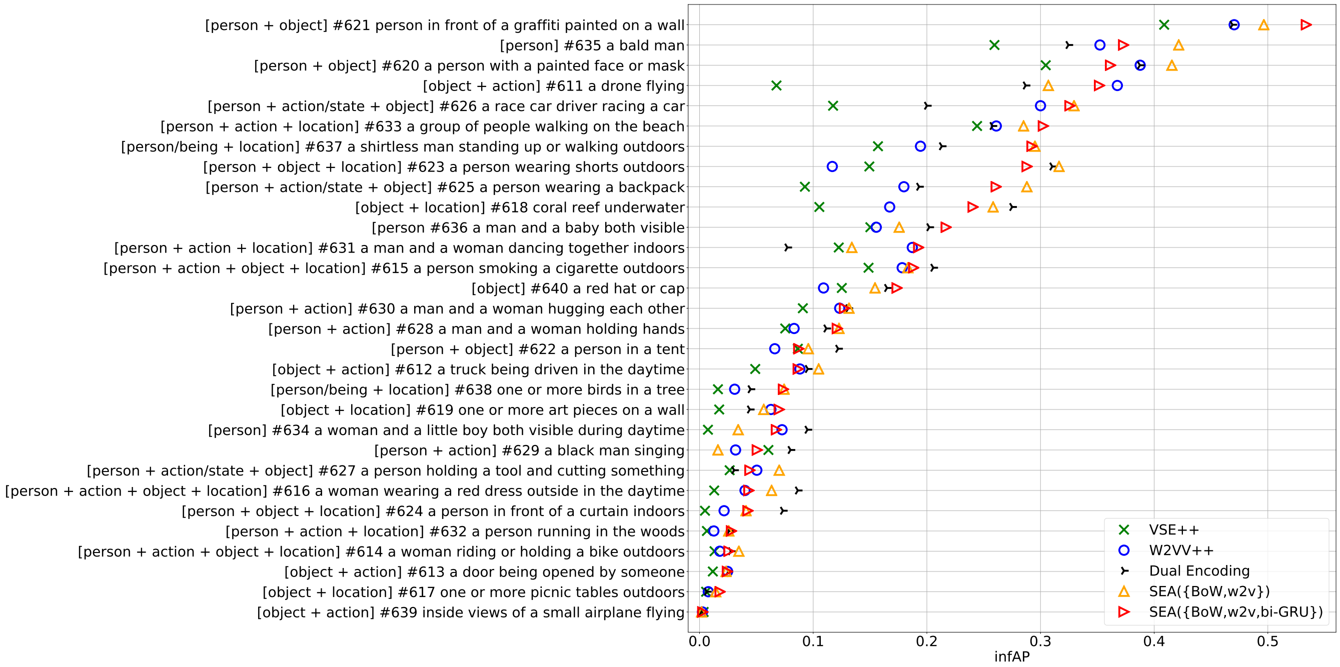


Fig. 6. **Performance of varied models in the TRECVID 2019 (TV19) AVS task**, which is to find amidst a set of one million unlabeled videos those relevant with respect to 30 test queries. For result analysis, each query is preceded by the TRECVID-specified query type, *e.g.,* [person + object] or [person + action + location], and ID. The queries are sorted in descending order in terms of their infAP scores by *SEA* ({BoW,w2v,bi-GRU}). As the key difference of the varied models is whether multiple sentence encoders are used and how they are combined, the leading performance of the *SEA* series verifies the effectiveness of the proposed method, namely multi-space network plus multi-loss training.

TABLE VI
STATE-OF-THE-ART ON **TRECVID AVS**. *SEA* SURPASSES THE PRIOR ART. LATE AVERAGE FUSION OF TWO *SEA* MODELS OR *SEA* ({BOW,W2V}) AND DUAL ENCODING BOOSTS THE PERFORMANCE.

Model	**TV16**	**TV17**	**TV18**	**TV19**	**MEAN**
Top-3 TRECVID finalists					
Rank 1	5.4 [25]	20.6 [26]	12.1 [35]	16.3 [13]	n.a.
Rank 2	5.1 [27]	15.9 [28]	8.7 [56]	16.0 [57]	n.a.
Rank 3	4.0 [58]	12.0 [29]	8.2 [59]	12.3 [60]	n.a.
VideoStory [20], [61]	8.7	15.0	n.a.	n.a.	n.a.
VSE++ [43]	13.5	16.3	10.6	9.8	12.6
W2VV++ [12]	16.2	22.3	10.1	13.9	15.6
Dual Encoding [14]	16.5	22.8	11.7	15.2	16.6
Extended Dual Encoding [24]	15.9	24.4	12.6	n.a.	n.a.
SEA({BoW,w2v})	15.7	23.4	**12.8**	16.6	17.1
SEA({BoW,w2v,bi-GRU})	16.4	22.8	12.5	16.7	17.1
SEA({BoW,w2v}) + *SEA*({BoW,w2v,bi-GRU})	16.6	23.5	12.6	17.2	17.5
SEA({BoW,w2v}) + Dual Encoding	**17.3**	**25.0**	**12.8**	**17.1**	**18.1**

which naturally reflects the state-of-the-art. We again compare with W2VV++, VSE++ and Dual Encoding, re-training them using the TRECVID setup as described in Section IV-A2. We also include VideoStory [20] which uses BoW as its sentence encoder, and Extended Dual Encoding [24], a very recent work which makes use of more than one encodings of the visual and textual content and two distinct attention mechanisms.

Results. The performance on the TRECVID test data is shown in Table VI. The proposed *SEA* model surpasses the prior art. While the Extended Dual Encoding network [24] appears to be on par with the SEA models, [24] has factually used the ground truth of the test set, which shall be unavailable in real applications, to select the best performing models. By contrast, our model selection is performed exclusively based on an independent validation set (see Table II), and thus more practical.

Late average fusion of Dual Encoding and *SEA* boosts the performance further, see the last row. Note that previous top-performing submissions boost their performance by late (average) fusion of a handful of models [13], [26], [35] or nearly hundred models [60]. In this context, the capability of *SEA* to advance the state-of-the-art with a single model is a big advantage for AVS at large-scale.

Fig. 6 shows how each model performs on the individual queries from the TV19 task, by searching over the one-million V3C1 collection. Each query is preceded by a TRECVID-specified query type that reflects the query complexity to some extent [5]. A query comprised of person, action, object and location tends to be more complex and thus more difficult to address than a query of person. While such a pattern can largely be observed from Fig. 6, exceptions are not uncommon. Consider query #639, for instance. Although the top-ranked videos show small airplane flying, they are mostly "external view", rather than "inside view" as required. Such a geometric property has not been effectively captured by the current sentence encoders that are fully data-driven. We consider the *SEA* model, with its flexibility to harvest new encoders, promising to attack the deficiency.

3) On TGIF and MSVD: For both datasets, we follow the data partition specified by their developers. That is, training / validation / test is 78,799 / 10,705 / 11,351 for TGIF and 1,200 / 100 / 670 for MSVD. All captions are used. The state-of-the-

TABLE VII
STATE-OF-THE-ART ON TGIF FOR TEXT-BASED VIDEO RETRIEVAL.

Model	R@1	R@5	R@10	Med r	mAP
HGR [30]	4.5	12.4	17.8	160	n.a.
Dual Encoding [14]	9.1	21.3	28.6	50	15.7
W2VV++ [12]	9.4	22.3	29.8	48	16.2
CF-GNN [31]	10.2	23.0	30.7	44	n.a.
SEA ({BoW,w2v,GRU})	10.2	23.6	31.3	41	17.2
SEA ({BoW,w2v,BERT})	10.7	24.4	31.9	37	17.9
SEA ({BoW,w2v,GRU,BERT})	**11.1**	**25.2**	32.7	36	18.4
SEA ({BoW,w2v,bi-GRU,BERT})	**11.1**	**25.2**	**32.8**	**35**	**18.5**

TABLE VIII
STATE-OF-THE-ART ON MSVD FOR TEXT-BASED VIDEO RETRIEVAL.

Model	R@1	R@5	R@10	Med r	mAP
Dual Encoding [14]	20.3	46.8	59.7	6	32.9
CF-GNN [31]	22.8	50.9	63.6	6	n.a.
W2VV++ [12]	22.4	51.6	64.8	5	36.1
SEA ({BoW,w2v,GRU})	23.2	52.9	66.2	5	37.2
SEA ({BoW,w2v,BERT})	**24.6**	**55.0**	**67.9**	**4**	**38.7**
SEA ({BoW,w2v,GRU,BERT})	24.4	54.1	67.6	5	38.3
SEA ({BoW,w2v,bi-GRU,BERT})	23.9	53.9	67.3	5	38.0

art following such a setting is HGR [30] and CF-GNN [31] on TGIF and CF-GNN on MSVD. Therefore, we compare with these two models. Dual Encoding and W2VV++ are also included.

Results. As shown in Table VII and Table VIII, our *SEA* model is again the best. Given that the amount of the training data in MSVD is substantially less than that of TGIF, the peak performance of *SEA* on MSVD is reached with less sentence encoders.

F. Efficiency Analysis

We report in Table IX the amount of trainable parameters, training time and inference time of the SEA models with varied setups on MSR-VTT, TGIF and MSVD. Two state-of-the-art methods, *i.e.,* Dual Encoding [14] and W2VV++ [12], are included as well. For a fair comparison, all models use the same size of 2,048 for their common spaces. For all models, the computational cost of video embedding is excluded from the inference time as this step is done once in an offline mode. Concerning the training time, Dual Encoding is slower than W2VV++ and SEA on MSR-VTT and MSVD, while faster on TGIF. In particular, *SEA* ({BoW,w2v,biGRU,BERT}) requires the longest training time of 4.9 hours on TGIF, as we find that the model needs more training epochs to trigger the early stop mechanism on this dataset.

For each model, its inference time to answer a given query consists of two parts: 1) query embedding that projects the query into a common space (for Dual Encoding and W2VV++) or multiple common spaces (for the *SEA* models), and 2) ranking that performs cross-modal matching between the query and all videos in a test set and sorting the videos accordingly. The main computational overhead is due to the online inference of the BERT encoder. Still, query embedding can be done within 19 milliseconds. As the cross-modal matching is executed in parallel on GPU, the ranking is extremely fast, costing around one millisecond. The inference time per query is around 20 milliseconds. Hence, our model is sufficiently fast to support real-time interactive video retrieval.

V. CONCLUSIONS

We have described a method for exploiting diverse sentence encoders for ad-hoc video search. Our experiments show the importance of building a query representation learning network that supports text-video matching in multiple encoder-specific common spaces. Nonetheless, the multi-space network architecture alone is inadequate. In order to effectively utilize complementaries among the individual common spaces, the network has to be end-to-end trained with a combined loss. On four benchmark datasets including MSR-VTT, TRECVID AVS 2016–2019, TGIF and MSVD, our proposed *SEA* model with multi-space multi-loss learning surpasses the prior art.

APPENDIX

The impact of the common space dimensionality. As shown in Table X, except for using a relatively small value of 256, the dimensionality of the common space has a marginal impact on the performance of the proposed method. We recommend to use 2,048 to strikes a proper balance between performance and model complexity.

The role of pretraining corpus. As noted in Section III-B, we use w2v and BERT which were pre-trained on Flickr tags [32] and web documents [41], respectively. To investigate if better performance can be obtained by pre-training the two encoders on the same corpus, we have re-trained w2v on Wikipedia dumps and book corpus as used for BERT. We do not try the opposite direction, *i.e.,* re-training BERT on the Flickr data, since Flickr tags are not natural-language text and thus unsuited for training BERT. As shown in Table XI, the Flickr version of w2v is slightly better.

REFERENCES

[1] F. Berns, L. Rossetto, K. Schoeffmann, C. Beecks, and G. Awad, "V3C1 dataset: An evaluation of content characteristics," in *ICMR*, 2019.

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

[3] G. Awad, A. Butt, J. Fiscus, D. Joy, A. Delgado, M. Michel, A. Smeaton, Y. Graham, G. J. F. Jones, W. Kraaij, G. Quénot, M. Eskevich, R. Ordelman, and B. Huet, "Trecvid 2017: Evaluating ad-hoc and instance video search, events detection, video captioning and hyperlinking," in *TRECVID Workshop*, 2017.

[4] G. Awad, A. Butt, K. Curtis, Y. Lee, J. Fiscus, A. Godil, D. Joy, A. Delgado, A. F. Smeaton, Y. Graham, W. Kraaij, G. Quénot, J. Magalhaes, D. Semedo, and S. Blasi, "Trecvid 2018: Benchmarking video activity detection, video captioning and matching, video storytelling linking and video search," in *TRECVID Workshop*, 2018.

[5] G. Awad, A. Butt, K. Curtis, Y. Lee, J. Fiscus, A. Godil, A. Delgado, A. F. Smeaton, Y. Graham, W. Kraaij, and G. Quénot, "Trecvid 2019: An evaluation campaign to benchmark video activity detection, video captioning and matching, and video search & retrieval," in *TRECVID Workshop*, 2019.

[6] M. Naphade, J. Smith, J. Tešić, S.-F. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis, "Large-scale concept ontology for multimedia," *IEEE Multimedia*, vol. 13, no. 3, pp. 86–91, 2006.

[7] A. Hauptmann, R. Yan, W.-H. Lin, M. Christel, and H. Wactlar, "Can high-level concepts fill the semantic gap in video retrieval? a case study with broadcast news," *T-MM*, vol. 9, no. 5, pp. 958–966, 2007.

TABLE IX

TRAINING AND INFERENCE TIME OF DUAL ENCODING, W2VV++ AND OUR PROPOSED SEA MODELS ON DISTINCT DATASETS. THE DIMENSIONALITY OF THE COMMON SPACE FOR ALL MODELS IS 2,048. EXPERIMENTS ARE DONE WITH PYTORCH (1.2.0) ON AN NVIDIA 1080TI GPU.

Model	**MSR-VTT**				**TGIF**				**MSVD**			
	#parameters (million)	*training time* (hr)	*query embedding* (ms)	*ranking* (ms)	*#parameters* (million)	*training time* (hr)	*query embedding* (ms)	*ranking* (ms)	*#parameters* (million)	*training time* (hr)	*query embedding* (ms)	*ranking* (ms)
Dual Encoding [14]	95.9	2.9	14.8	0.5	86.5	2.2	16.6	0.9	83.7	0.7	13.7	0.4
W2VV++ [12]	35.8	1.2	2.1	0.5	26.4	2.4	2.2	0.9	23.7	0.3	1.8	0.4
SEA(BoW,w2v)	33.5	0.9	1.2	0.6	26.0	3.1	1.0	1.0	23.8	0.1	0.9	0.6
SEA(BoW,w2v,GRU)	52.6	2.7	2.4	0.8	43.2	4.4	2.4	1.1	40.5	0.2	2.2	0.8
SEA(BoW,w2v,bi-GRU)	59.4	2.4	2.8	0.8	50.0	4.9	3.0	1.1	47.3	0.4	2.6	0.8
SEA(BoW,w2v,BERT)	43.5	1.0	15.8	0.8	35.9	3.6	15.9	1.1	33.8	0.2	15.5	0.8
SEA(BoW,w2v,GRU,BERT)	62.6	2.0	17.4	0.9	53.2	4.4	17.1	1.2	50.4	0.3	17.1	1.0
SEA(BoW,w2v,bi-GRU,BERT)	69.4	2.5	18.4	0.9	59.9	4.9	18.2	1.2	57.2	0.3	17.4	0.9

TABLE X

THE INFLUENCE OF THE COMMON SPACE DIMENSIONALITY $d_{c,i}$ ON THE MODEL PERFORMANCE. WE EVALUATE *SEA* ({BOW,W2V,GRU}) ON MSR-VTT.

$d_{c,i}$	**R@1**	**R@5**	**R@10**	**Med r**	**mAP**
256	11.0	29.5	40.4	18	20.4
512	12.0	31.5	42.7	16	21.9
1,024	12.1	31.6	42.9	**15**	22.0
2,048	**12.2**	**31.9**	**43.1**	**15**	**22.1**
4,096	**12.2**	31.8	**43.1**	**15**	**22.1**
8,192	12.1	31.4	42.6	16	21.8

TABLE XI

PERFORMANCE OF SEA WITH W2V PRE-TRAINED ON DISTINCT CORPUS. WE EVALUATE *SEA* ({BOW,W2V,BI-GRU,BERT}) ON MSR-VTT.

Corpus for w2v	**R@1**	**R@5**	**R@10**	**Med r**	**mAP**
Flickr tags	13.1	33.4	45.0	14	23.3
Wiki & book corpus	13.0	33.4	44.9	14	23.3

[8] C. G. M. Snoek and M. Worring, "Concept-based video retrieval," *Foundations and Trends in Information Retrieval*, vol. 2, no. 4, pp. 215–322, 2009.

[9] Y.-J. Lu, H. Zhang, M. de Boer, and C.-W. Ngo, "Event detection with zero example: Select the right and suppress the wrong concepts," in *ICMR*, 2016.

[10] F. Markatopoulou, D. Galanopoulos, V. Mezaris, and I. Patras, "Query and keyframe representations for ad-hoc video search," in *ICMR*, 2017.

[11] G. Awad, D.-D. Le, C.-W. Ngo, V.-T. Nguyen, G. Quénot, C. Snoek, and S. Satoh, "Video indexing, search, detection, and description with focus on trecvid," in *ICMR*, 2017.

[12] X. Li, C. Xu, G. Yang, Z. Chen, and J. Dong, "W2VV++: Fully deep learning for ad-hoc video search," in *ACMMM*, 2019.

[13] X. Wu, D. Chen, Y. He, H. Xue, M. Song, and F. Mao, "Hybrid sequence encoder for text based video retrieval," in *TRECVID Workshop*, 2019.

[14] J. Dong, X. Li, C. Xu, S. Ji, Y. He, G. Yang, and X. Wang, "Dual encoding for zero-example video retrieval," in *CVPR*, 2019.

[15] N. Mithun, J. Li, F. Metze, and A. K. Roy-Chowdhury, "Learning joint embedding with multimodal cues for cross-modal video-text retrieval," in *ICMR*, 2018.

[16] Y. Liu, S. Albanie, A. Nagrani, and A. Zisserman, "Use what you have: Video retrieval using representations from collaborative experts," in *BMVC*, 2019.

[17] H. Luo, L. Ji, B. Shi, H. Huang, N. Duan, T. Li, X. Chen, and M. Zhou, "UniViLM: A unified video and language pre-training model for multimodal understanding and generation," *arXiv preprint arXiv:2002.06353*, 2020.

[18] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, "Howto100M: Learning a text-video embedding by watching hundred million narrated video clips," in *ICCV*, 2019.

[19] Y. Pan, T. Mei, T. Yao, . Li, and Y. Rui, "Jointly modeling embedding and translation to bridge video and language," in *CVPR*, 2016.

[20] A. Habibian, T. Mensink, and C. G. M. Snoek, "Video2vec embeddings recognize events when examples are scarce," *T-PAMI*, vol. 39, no. 10, pp. 2089–2103, 2017.

[21] M. Wray, D. Larlus, G. Csurka, and D. Damen, "Fine-grained action retrieval through multiple parts-of-speech embeddings," in *ICCV*, 2019.

[22] B. Zhang, H. Hu, and F. Sha, "Cross-modal and hierarchical modeling of video and text," in *ECCV*, 2018.

[23] J. Lokoć, T. Soućek, P. Veselý, F. Mejzlík, J. Ji, C. Xu, and X. Li, "A W2VV++ case study with automated and interactive text-to-video retrieval," in *ACMMM*, 2020.

[24] D. Galanopoulos and V. Mezaris, "Attention mechanisms, signal encodings and fusion strategies for improved ad-hoc video search with dual encoding networks," in *ICMR*, 2020.

[25] D.-D. Le, S. Phan, V.-T. Nguyen, B. Renoust, T. Nguyen, V.-N. Hoang, T. Ngo, M.-T. Tran, Y. Watanabe, M. Klinkigt *et al.*, "NII-HITACHI-UIT at TRECVID 2016," in *TRECVID Workshop*, 2016.

[26] C. G. M. Snoek, X. Li, C. Xu, and D. C. Koelma, "University of Amsterdam and Renmin university at TRECVID 2017: Searching video, detecting events and describing video," in *TRECVID Workshop*, 2017.

[27] F. Markatopoulou, A. Moumtzidou, D. Galanopoulos, T. Mironidis, V. Kaltsa, A. Ioannidou, S. Symeonidis, K. Avgerinakis, S. Andreadis *et al.*, "ITI-CERTH participation in TRECVID 2016," in *TRECVID Workshop*, 2016.

[28] K. Ueki, K. Hirakawa, K. Kikuchi, T. Ogawa, and T. Kobayashi, "Waseda_Meisei at TRECVID 2017: Ad-hoc video search," in *TRECVID Workshop*, 2017.

[29] P. Nguyen, Q. Li, Z.-Q. Cheng, Y.-J. Lu, H. Zhang, X. Wu, and C.-W. Ngo, "VIREO@TRECVID 2017: Video-to-text, ad-hoc video search and video hyperlinking," in *TRECVID Workshop*, 2017.

[30] S. Chen, Y. Zhao, Q. Jin, and Q. Wu, "Fine-grained video-text retrieval with hierarchical graph reasoning," in *CVPR*, 2020.

[31] W. Wang, J. Gao, X. Yang, and C. Xu, "Learning coarse-to-fine graph neural networks for video-text retrieval," *T-MM*, 2020, in press.

[32] J. Dong, X. Li, and C. G. M. Snoek, "Predicting visual features from text for image and video caption retrieval," *T-MM*, vol. 20, no. 12, pp. 3377–3388, 2018.

[33] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," *T-PAMI*, vol. 40, no. 6, pp. 1437–1451, 2018.

[34] Y. Yu, J. Kim, and G. Kim, "A joint sequence fusion model for video question answering and retrieval," in *ECCV*, 2018.

[35] X. Li, J. Dong, C. Xu, J. Cao, X. Wang, and G. Yang, "Renmin University of China and Zhejiang Gongshang University at TRECVID 2018: Deep Cross-Modal Embeddings for Video-Text Retrieval," in *TRECVID Workshop*, 2018.

[36] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Heidelberg: Springer-Verlag, 2006.

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

[38] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *EMNLP*, 2014.

[39] 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.

[40] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[41] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2019.

[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.

[43] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler, "VSE++: Improving visual-semantic embeddings with hard negatives," in *BMVC*, 2018.

[44] X. Yang, J. Dong, Y. Cao, X. Wang, M. Wang, and T.-S. Chua, "Tree-augmented cross-modal encoding for complex-query video retrieval," in *SIGIR*, 2020.

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

[46] 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.

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

[48] L. Rossetto, H. Schuldt, G. Awad, and A. A. Butt, "V3c – a research video collection," in *MMM*, 2019.

[49] E. Yilmaz and J. A. Aslam, "Estimating average precision with incomplete and imperfect judgments," in *CIKM*, 2006.

[50] A. Joulin, L. van der Maaten, A. Jabri, and N. Vasilache, "Learning visual features from large weakly supervised data," in *ECCV*, 2016.

[51] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019.

[52] M. Kratochvíl, P. Veselý, F. Mejzlík, and J. Lokoč, "SOM-Hunter: Video browsing with relevance-to-som feedback loop," in *MMM*, 2020.

[53] J. Lokoč, W. Bailer, K. Schoeffmann, B. Muenzer, and G. Awad, "On influential trends in interactive video retrieval: Video browser showdown 2015–2017," *T-MM*, vol. 20, no. 12, pp. 3361–3376, 2018.

[54] L. Rossetto, R. Gasser, J. Lokoc, W. Bailer, K. Schoeffmann, B. Muenzer, T. Soucek, P. A. Nguyen, P. Bolettieri, A. Leibetseder, and S. Vrochidis, "Interactive video retrieval in the age of deep learning - detailed evaluation of VBS 2019," *T-MM*, 2020, in press.

[55] N. Mithun, J. Li, F. Metze, and A. K. Roy-Chowdhury, "Joint embeddings with multimodal cues for video-text retrieval," *International Journal of Multimedia Information Retrieval*, vol. 8, no. 1, pp. 3–18, 2019.

[56] P.-Y. Huang, J. Liang, V. Vaibhav, X. Chang, and A. Hauptmann, "Informedia@TRECVID 2018: Ad-hoc video search with discrete and continuous representations," in *TRECVID Workshop*, 2018.

[57] X. Li, J. Ye, C. Xu, S. Yun, L. Zhang, X. Wang, R. Qian, and J. Dong, "Renmin University of China and Zhejiang Gongshang University at TRECVID 2019: Learn to search and describe videos," in *TRECVID Workshop*, 2019.

[58] J. Liang, J. Chen, P. Huang, X. Li, L. Jiang, Z. Lan, P. Pan, H. Fan, Q. Jin, J. Sun *et al.*, "Informedia @ Trecvid 2016," in *TRECVID Workshop*, 2016.

[59] M. Bastan, X. Shi, J. Gu, Z. Heng, C. Zhuo, D. Sng, and A. Kot, "NTU ROSE lab at TRECVID 2018: Ad-hoc video search and video to text," in *TRECVID Workshop*, 2018.

[60] K. Ueki, T. Hori, and T. Kobayashi, "Waseda_Meisei_SoftBank at TRECVID 2019: Ad-hoc video search," in *TRECVID Workshop*, 2019.

[61] D. C. Koelma and C. G. M. Snoek, "Query understanding is key for zero-example video search," in *TRECVID Workshop*, 2017.

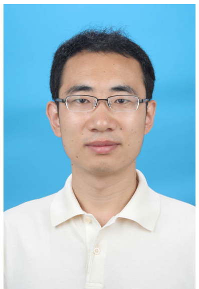
Xirong Li received the B.S. and M.E. degrees from Tsinghua University, Beijing, China, in 2005 and 2007, respectively, and the Ph.D. degree from the University of Amsterdam, Amsterdam, The Netherlands, in 2012, all in computer science. He is currently an Associate Professor with the Key Lab of Data Engineering and Knowledge Engineering, Renmin University of China, Beijing, China. His research is multimedia intelligence.

Dr. Li was recipient of the ACMMM 2016 Grand Challenge Award, the ACM SIGMM Best Ph.D. Thesis Award 2013, the IEEE TRANSACTIONS ON MULTIMEDIA Prize Paper Award 2012, and the Best Paper Award of ACM CIVR 2010. He served as program co-chair of Multimedia Modeling 2021. He is associate editor of ACM TOMM and the Multimedia Systems journal.

Fangming Zhou received his B.S. degree in Nano Materials in Nanjing University of Science and Technology, Nanjing, China in 2019. He is currently a graduate student at School of Information, Renmin University of China, pursuing his master degree on multimedia retrieval.

Chaoxi Xu received his B.S. and M.E. degrees in Computer Science from Renmin University of China, Beijing, China, in 2017 and 2020, respectively. He is currently an assistant engineer at the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China.

Jiaqi Ji received his B.S degree in Software Engineering from Taiyuan University of Technology, Taiyuan, China in 2019. He is currently a graduate student at the School of Information, Renmin University of china, pursuing his master degree on video retrieval.



Gang Yang received his Ph.D. degree in Innovative Life Science from University of Toyama, Toyama, Japan in 2009. He is currently an Associate Professor at School of Information, Renmin University of China, Beijing, China. His research interests include computational intelligence, multimedia computing and machine learning.