Video and Text Matching with Conditioned Embeddings

Ameen Ali
Tel Aviv University

Idan Schwartz
Technion
NetApp

Tamir Hazan
Technion

Lior Wolf
Tel Aviv University

Abstract

We present a method for matching a text sentence from a given corpus to a given video clip and vice versa. Traditionally video and text matching is done by learning a shared embedding space and the encoding of one modality is independent of the other. In this work, we encode the dataset data in a way that takes into account the query's relevant information. The power of the method is demonstrated to arise from pooling the interaction data between words and frames. Since the encoding of the video clip depends on the sentence compared to it, the representation needs to be recomputed for each potential match. To this end, we propose an efficient shallow neural network. Its training employs a hierarchical triplet loss that is extendable to paragraph/video matching. The method is simple, provides explainability, and achieves state-of-the-art results for both sentence-clip and video-text by a sizable margin across five different datasets: ActivityNet, DiDeMo, YouCook2, MSR-VTT, and LSMDC. We also show that our conditioned representation can be transferred to video-guided machine translation, where we improved the current results on VATEX. Source code is available at `https://github.com/AmeenAli/VideoMatch`.

1. Introduction

The process of matching data from two domains is often seen as a sequential process of comparison between two static representations, obtained by preprocessing these domains. However, it is possible that a more effective way is to consider recent models for cognitive search, in which the content of the query signal from the first domain changes our perception of the data from the second domain.

In this work, we consider hierarchical representations of two domains, in which repeated interaction between the two domains modifies the representations of each data source. In particular, we consider the problem of matching multi-clip video to multi-sentence text. The interaction-based representations are a form of attention. As seen in Fig. 1, when matching the sentence "fry onions until golden then add carrots and fry for 5 mins" to a cooking video segment,
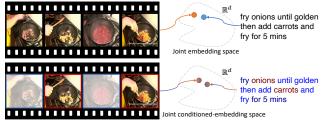

Figure 1. Current approaches use a joint embedding space for efficient video and text retrieval via a dot-product (illustrated in the top row). However, retrieving the relevant video to a given text query requires a concise representation that consists of only the relevant frames. For this reason, we propose a new conditioned embedding that encodes all the videos conditioned on a given query. On the bottom row, we highlight with red color the relevant words given the video clip. We also mark with a red border the relevant frames given the text query. Note the query is "fry onions ... add carrots ...", and the selected relevant frames show onion and carrots, while the non-relevant frame shows meat.

the frames showing onions, or carrots are highlighted in the video representation. Focusing on the video and text domains enables us to benefit from the extensive literature on image and text embedding, as well as from the existence of multiple existing benchmarks.

We represent the video frames with a pre-trained CNN and the words using the GloVe [31] representation, both of which are standard. Then, a temporal context is added to the video by applying a unidirectional GRU [6]. This is done at the clip level for video and the sentence level for the text. Given a video clip and a possibly-matching sentence, we compute an affinity score between every frame (clip) and word (sentence) by projecting the domain-specific representations to a shared vector space, in which the cosine similarity is employed.

The individual similarities are aggregated to define per frame (per word) weights, which are used to obtain one weighted-sum representation of the video clip (sentence). The video clips and sentences are then each processed by another pair of GRUs to create another layer of representation, whose initial state depends on information collected from the entire video and text paragraph. Max pooling is

then employed to obtain a single vector representing each video (all clips) and each paragraph (all sentences). These representations are projected to a joint vector space, in which they are compared.

Since the encoding of the video clip depends on the sentence it was compared to, the representation needs to be recomputed for each potential match. While this does not affect the asymptotic complexity of the search process, it does require applying the encoding mechanism N^2 times, instead of $2N$ times, where N is the number of videos. However, in practice, we had no challenge running on the existing benchmarks, using a shortlist obtained using static embeddings.

The new method is able to outperform by a sizable margin the state of the art results for the cross-modality text/video retrieval on a diverse set of video to text matching benchmarks, including ActivityNet [17], DiDeMo [13], YouCook2 [51] , Vatex [39] , MSR-VTT [43] and LSMDC [32]. The robustness of our method is further demonstrated by its application to the task of video-guided based translation on Vatex, surpassing the recent baseline.

2. Related Work

In the task of **video/text retrieval**, given a set of video descriptions, we rank them according to video relevancy. Early works clustered videos and captions by considering metadata, video tags, and video features(*i.e.*, SIFT, BoW) [1, 40, 14]. Afterward, a model that finetunes video caption generation model was proposed [17]. However, a dedicated matching model significantly improved the results [49]. We follow the same strategy but improve in two aspects: (i) we condition the video and paragraph embeddings, and (ii) we use a weighted interaction score, instead of a dot product. The video-paragraph retrieval relies on dense clip annotations, which are not always available. Therefore, instead, recent methods focus on the sentence-video retrieval variant. Liu *et al.* [21] utilize different experts such as speech, audio, motion, OCR, appearance, and face detection. Gabeur *et al.* [10], improved this approach by leveraging transformers networks. Our approach is much more straightforward and considers only the video and the text. Lei *et al.* [18] suggested the use of just a few sparsely sampled clips for matching. Patrick *et al.* [30] propose incorporating a generator and a contrastive loss to consider semantically-related negative samples better. Different from all the methods mentioned above, we are the first to study the power of conditioned video/clip embedding, which results in a significant performance improvement. Recently, the HowTo100M dataset has been used as a basis for training large-scale approaches [25, 52, 23]. Notably, out approach remains competitive even when compared to pre-trained approaches, despite the significant increase in data pre-training methods use.

In **temporal grounding**, where the task is to localize a segment in the video that semantically corresponds to the given sentence. Most of the existing approaches to this problem use dense supervision of event locations for training [13, 11, 41, 4, 41, 4, 48, 50]. These annotations are not always available and recent methods, such as Mithun *et al.* focus on weakly-supervised temporal grounding by leveraging text-to-frame attention scores [36, 27, 12, 8]. The most relevant to our approach, Yuan *et al.* [48] uses co-attention for video localization. However, our co-attention is simpler as it works in a parallel manner, while Yuan *et al.* infers attention in an alternating manner, *i.e.*, the attention is computed multiple times sequentially.

Hierarchical encoding deals with long text (*i.e.*, paragraph) or a long video with multiple clips. Li *et al.* [19] propose encoding a paragraph with a hierarchical autoencoder. Following, a hierarchical RNN was proposed [29, 46]. Niu *et al.* [28] leverage the hierarchy of a sentence parse tree to match a sentence with an image. Chen *et al.* [5] proposed using hierarchical graph encoding on the local and global level of the data. Similar to our approach, Zhang *et al.* [49] embed video and paragraph in a two-level hierarchical manner. The first level embeds the clips and sentences, followed by a second level that embeds the video and paragraph. However, the embeddings are independent. In this work, we demonstrate that conditioning the clip embedding on the matched sentence, and vice versa significantly improves the performance.

Multimodal Attention has been a prominent tool for modeling the interaction between the two domains, since it models interactions in a way that selects important frames and words. We follow the same practice and use efficient interaction-based attention as a tool for conditioning. In early work, image attention was proposed for improved image captioning [44]. This idea was later extended to visual question answering [42]. Other approaches improved the vector-fusion operator for better interaction modeling [9, 16, 2]. Some approaches imitate a multi-step reasoning with stacked attention modules sequentially [45]. Lu *et al.* [22] proposed a co-attention module that attends to visual and textual modalities jointly. A more advanced approach for co-attention used a bilinear module that efficiently generates attention for every pair [15]. Next, a general factor-based framework that allows joint attention to any number of modalities was proposed [33, 35]. In recent work, the factor graph attention is used to allow interactions between video frames for video dialog [34], list of images for visual storytelling [3], and spatial information for navigation [24].

3. Method

Learning a similarity measure between text-to-video allows an efficient video retrieval of a video given text. In the following, we propose to learn from training data a similarity score between a video $v \in V$, and a paragraph $p \in P$. The training dataset $\mathcal{S} = \{(v_1, p_1), \ldots, (v_n, p_n)\}$ consists of n video samples and matching paragraphs. Each video sample $v_i \in V$ is a sequence of clips $v_i = \{c_1^i, \ldots, c_{n_i}^i\}$

and each clip j is composed of frames $\boldsymbol{c_j^i} = \{f_1^{ij}, \ldots, f_{n_{ij}}^{ij}\}$. The matching paragraph of a video/paragraph pair $(\boldsymbol{v_i}, \boldsymbol{p_i})$ consists of a sequence of sentences $\boldsymbol{p_i} = \{s_1^i, \ldots, s_{n_i}^i\}$, while for any $j = 1, \ldots, n_i$ the sentence $\boldsymbol{s_j^i}$ corresponds to the clip $\boldsymbol{c_j^i}$. The sentence itself is composed of words $\boldsymbol{s_j^i} = \{w_1^{ij}, \ldots, w_{m_{ij}}^{ij}\}$ where m_{ij} is the number of words in the j-th sentence $\boldsymbol{s_j^i}$ of paragraph $\boldsymbol{p_i}$. We note that although the j-th clip and the j-th sentence, of the a matching video/paragraph pair of any index i, are matched, their words and frames do not match, and the number of frames in a clip n_{ij} often differ from the number of words m_{ij} in its corresponding sentence.

Our training procedure is learning the similarity of paired video and paragraph $(\boldsymbol{v_i}, \boldsymbol{p_i}) \in \mathcal{S}$ by embedding both $\boldsymbol{v_i}$ and $\boldsymbol{p_i}$ in a joint space $\mathcal{R}^{d_{vp}}$. Each such training pair should lie close together in the joint space, while embedding a paragraph and a video that do not match should lie far apart. The embedding of the video and paragraph are dependent on each other. To search for the matching paragraph of a given video, we consider the pairwise Euclidean distances between the embedding of the video given each possible paragraph and the embedding of each paragraph given the video. The same process, in the other direction, is used for searching for the matching video given a paragraph.

3.1. Joint Clip and Sentence Encoding

The dataset $\mathcal{S} = \{(\boldsymbol{v_i}, \boldsymbol{p_i})\}_{i=1}^n$ contains paired videos and corresponding paragraphs that are describing the videos. These are embedded in a bottom-up manner that matches the hierarchical structure of the data. In the video pathway, the frames f_k^{ij} of the clip $\boldsymbol{c_j^i} = \{f_1^{ij}, \ldots, f_{n_{ij}}^{ij}\}$ are encoded using a pre-trained CNN, followed by a GRU-type RNN GRU_c which captures the temporal context of the n_{ij} frames of video clip $\boldsymbol{c_j^i}$:

$$(\bar{f}_1^{ij}, \ldots, \bar{f}_{n_{ij}}^{ij}) = \mathrm{GRU}_c\left(\mathrm{CNN}(f_1^{ij}), \ldots, \mathrm{CNN}(f_{n_{ij}}^{ij})\right). \tag{1}$$

Similarly, the m_{ij} words w_k^{ij} of the sentence $\boldsymbol{s_j^i} = \{w_1^{ij}, \ldots, w_{m_{ij}}^{ij}\}$ are encoded by GloVe [31] followed by an RNN

$$(\bar{w}_1^{ij}, \ldots, \bar{w}_{m_{ij}}^{ij}) = \mathrm{GRU}_s\left(\mathrm{GloVe}(w_1^{ij}), \ldots, \mathrm{GloVe}(w_{m_{ij}}^{ij})\right), \tag{2}$$

where GRU_s is the sentence RNN whose sequence is composed of words. Both GRUs have a single layer of GRU cells and an output of size d_e.

Up until this stage, the video and text pathways have been independent. Next, we apply our conditioning mechanism in order to highlight individual frames that are related to the words of the sentence and vice versa as illustrated in Fig. 2. For this purpose, we define the following matrix of interaction scores, which is given as a normalized dot product:

$$\mathrm{I}_{kk'}^{ii'jj'} = \left(\frac{A\bar{f}_k^{ij}}{\left\|A\bar{f}_k^{ij}\right\|}\right)^\top \left(\frac{B\bar{w}_{k'}^{i'j'}}{\left\|B\bar{w}_{k'}^{i'j'}\right\|}\right), \tag{3}$$

where $\mathrm{I} \in \mathbb{R}^{n_{ij} \times m_{i'j'}}$ and $A \in \mathbb{R}^{d_e \times d_e}, B \in \mathbb{R}^{d_e \times d_e}$ are learned matrices that re-embed the frames and the words to their joint space $\mathcal{R}^{d_e}$. The indices of the video/clip/frame i, j, k are distinguished from those of the paragraph/sentence/word i', j', k'.

We define the marginal interaction potential over frames and words to be

$\rho_c(k|\boldsymbol{c_j^i}, s_{j'}^{i'}) = \sum_{k'} \mathrm{I}_{kk'}^{ii'jj'}, \rho_s(k'|\boldsymbol{c_j^i}, s_{j'}^{i'}) = \sum_k \mathrm{I}_{kk'}^{ii'jj'}$.

The potentials are then transformed, using softmax, to a conditional probability distribution over frames and words given their clip and sentence.

$$\mu_c(k|\boldsymbol{c_j^i}, s_{j'}^{i'}) \propto \exp(\rho_c(k|\boldsymbol{c_j^i}, s_{j'}^{i'})),$$
$$\mu_s(k'|\boldsymbol{c_j^i}, s_{j'}^{i'}) \propto \exp(\rho_s(k'|\boldsymbol{c_j^i}, s_{j'}^{i'})). \tag{4}$$

With these marginal probabilities, we entangle the clips and the sentences by reducing their representation, namely, summarize the clip representation by combining its sentence-related frames and the sentence representation by its clip-related words:

$$c_{jj'}^{ii'} = \sum_{k=1}^{m_{ij}} \mu_c(k|\boldsymbol{c_j^i}, s_{j'}^{i'})\bar{f}_k^{ij},$$
$$s_{j'j}^{i'i} = \sum_{k'=1}^{n_{i'j'}} \mu_s(k'|\boldsymbol{c_j^i}, s_{j'}^{i'})\bar{w}_{k'}^{i'j'}. \tag{5}$$

Since these representations are derived from the joint interactions space of clips and sentences, the resulting embeddings of $c_{jj'}^{ii'}$ and $s_{jj'}^{ii'}$ are conditioned on each other. The next section describes an additional encoding hierarchy that enables video and paragraph matching.

3.2. Joint Video and Paragraph Encoding

We find it easy to extend out conditioned representation to the matching of a video and a paragraph. Given interaction-conditioned representation. *I.e.*, video's clips $\{c_{11}^{ii'}, \ldots, c_{n_{ii'}n_{ri}}^{ii'}\}$ and paragraph's sentences $\{s_{11}^{i'i}, \ldots, s_{n_{i'i}n_{ii'}}^{i'i}\}$. We encode them with a pair of GRUs where $n_{ii'} = \max(n_{i'}, n_i)$

$$(\hat{c}_1^{ii'}, \ldots, \hat{c}_{n_{ii'}}^{ii'}) = \mathrm{GRU}_v\left((c_{11}^{ii'}, \ldots, c_{n_{ii'}}^{ii'})\right),$$
$$(\hat{s}_1^{i'i}, \ldots, \hat{s}_{n_{ii'}}^{i'i}) = \mathrm{GRU}_p\left((s_{11}^{i'i}, \ldots, s_{n_{i'i}n_{ii'}}^{i'i})\right). \tag{6}$$

In case the number of clips and the number of sentences are not equal, we pad with a zero vector the missing representations.
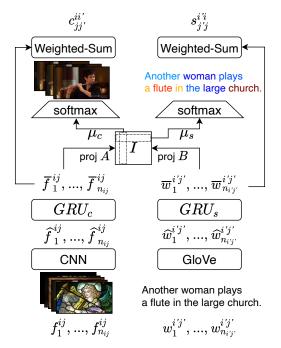
Figure 2. An overview of our first hierarchy for clip and sentence embedding computation. The frames f_k^{ij} of the video $\boldsymbol{v_i}$ and the clip $\boldsymbol{c_j^i}$ and the words $w_k^{i'j'}$ of the paragraph $\boldsymbol{p_{i'}}$ and the sentence $s_{j'}^{i'}$, are processed in the context of each other. A similarity matrix I is computed based on a dot-product. We calculate interaction potential scores μ_c and μ_s by marginalization. We then produce the conditioned embeddings $c_{jj'}^{ii'}$ and $s_{j'j}^{i'i}$ based on the interaction potential score.

Notably, to this point, each clip and sentence interact locally. In other words, objects in a clip only interact with words in the corresponding sentence and vice versa. However, we find it beneficial to also condition the video and paragraph in a global context, *i.e.*, words/objects interact with non-corresponding sentence/clip. To this end, we sample n_f frames from the entire video, along with the words of the entire paragraph. We then apply the same process as in Eq. 3 - Eq. 5, treating the sampled frames and words as a single clip and a single sentence. The only difference is that we employ a different set of matrices, instead of A and B (Eq. 3), which we denote by $A_0 \in \mathbb{R}^{d_e \times d_e}$ and $B_0 \in \mathbb{R}^{d_e \times d_e}$. We use the interaction-conditioned global video embeddings, denoted as $v_{ii'}^0$ as the RNN GRU_v's initial state. Similarly, we denote the global paragraph embedding $p_{ii'}^0$, which is used to initialize the RNN GRU_p.

Finally, to co-dependently embed the entire video $\boldsymbol{v_i}$ and the entire paragraph $\boldsymbol{p_i}$, max-pooling is applied across all video clips and paragraph sentences, respectively:

$$
\begin{aligned}
v_{ii'} &= \text{MaxPool}\left(\hat{c}_1^{ii'}, \ldots, \hat{c}_{n_{ii'}}^{ii'}\right), \\
p_{i'i} &= \text{MaxPool}\left(\hat{s}_1^{i'i}, \ldots, \hat{s}_{n_{ii'}}^{i'i}\right).
\end{aligned}
\tag{7}
$$

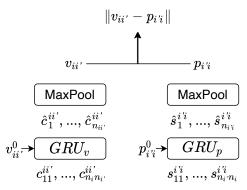The paragraph-level architecture is summarized in Fig. 3.3.

Figure 3. Our second hierarchy for the embedding and then matching a video and a paragraph based on the interaction-conditioned representations of the video's clips and the paragraph's sentences. Note that we employ this hierarchy only in the case of paragraph and video matching.

3.3. Training

The loss employed is based on the triplet loss of the video and paragraph embeddings together with an auxiliary loss that also takes the form of a triplet loss. Recall that for matching videos and paragraphs $(\boldsymbol{v_i}, \boldsymbol{p_i})$, the j-th clip $\boldsymbol{c_j^i}$ matches the j-th sentence $\boldsymbol{s_j^i}$. For this purpose, we learn another pair of matrices U and V to define a match score:

$$
\text{M}(x, y) = \left(\frac{Ux}{\|Ux\|}\right)^{\top}\left(\frac{Vy}{\|Vy\|}\right).
\tag{8}
$$

We note that for the video/paragraph case, U and V are only used during training and do not play a role in matching in inference time. In total, we learn three normalized dot-product cross-domain interactions. In Eq. 3 we learn A and B at the frame and word level. Similarly, we learn A_0 and B_0 for matching frames sampled from entire-video and words sampled from the entire-paragraph. Last, we learn U and V at the clip and sentence level.

In this metric learning task, positive pairs are obtained by co-embedding corresponding video and paragraphs (index i) using the corresponding clip/sentence (index j). Negative pairs are obtained using either unmatching clips and sentences, introducing a second index $j' \neq j$, or using entirely different video and paragraph, denoted by the index $i' \neq i$. The mini-batches consider b pairs of matching videos and paragraphs. Then, all non-matching pairs in the mini-batch are considered as negative samples, *i.e.*,

$$
\mathcal{L} = \sum_{i=1}^{b}\sum_{i' \neq i} \text{ReLU}(\|v_{ii} - p_{ii}\| - \|v_{ii'} - p_{i'i}\| + \alpha_1)
$$

$$
+ \sum_{i=1}^{b}\sum_{j=1}^{n_i}\sum_{j' \neq j} \text{ReLU}(\text{M}(c_{jj}^{ii}, s_{jj}^{ii}) - \text{M}(c_{jj'}^{ii}, s_{j'j}^{ii}) + \alpha_2)
$$

$$
+ \sum_{i=1}^{b}\sum_{i' \neq i}\sum_{j=1}^{n_i}\sum_{j'=1}^{n_{i'}} \text{ReLU}(\text{M}(c_{jj}^{ii}, s_{jj}^{ii}) - \text{M}(c_{jj'}^{ii'}, s_{j'j}^{ii'}) + \alpha_2),
\tag{9}
$$

Dataset	Method	Text2Vid			Vid2Text		
		R@1	R@5	MdR	R@1	R@5	MdR
ActivityNet	MMT‡ [10]	28.7%	61.4%	3.3	28.9%	61.1%	4.0
	FSE [49]	18.2%	44.8%	7.0	16.7%	43.1%	7.0
	HSE [49]	20.5%	49.3%	-	18.7%	48.1%	-
	CE‡ [21]	18.2%	47.7%	6.0	17.7%	46.6%	6.0
	MMT‡ [10]	22.7%	54.2%	5.0	22.9%	54.8%	4.3
	clipBERT†* [18]	21.3%	49.0%	-	-	-	-
	Ours	**25.4%**	**59.1%**	**3.0**	**26.1%**	**60.0%**	**3.0**
LSMDC	MMT‡ [10]	12.9%	29.9%	19.3	12.3%	28.6%	20.0
	HT100 [25]	7.1%	19.6%	40.0	-	-	-
	JFusion [47]	9.1%	21.2%	36.0	-	-	-
	HSE [49]	9.5%	25.8%	23.0	9.7%	25.1%	23.0
	MMT‡ [10]	13.2%	29.2%	21.0	12.1%	29.3%	22.5
	Ours	**14.9%**	**33.2%**	**19.0**	**15.3%**	**34.1%**	**18.0**
MSR-VTT	HT [25]	14.9%	40.2%	9.0	-	-	-
	MMT‡ [10]	26.6%	57.1%	4.0	27.0%	57.5%	3.7
	ActBERT [52]	16.3%	42.8%	10.0	-	-	-
	HERO [20]	16.8%	43.4%	-	-	-	-
	HERO† [20]	20.5%	47.6%	-	-	-	-
	Dual Encoding [7]	7.7%	22.0%	32.0	13.0%	30.8%	15.0
	HGR [5]	9.2%	26.2%	24.0	15.0%	36.7%	11.0
	JSFusion [47]	10.0%	31.2%	13.0	-	-	-
	Cues [26]	12.5%	32.1%	16.0	7.0%	20.9%	38.0
	HSE [49]	17.9%	45.0%	23.0	18.5%	45.2%	23.0
	MMT‡ [10]	24.6%	54.0%	4.0	21.1%	49.4%	6.0
	clipBERT†* [18]	22.0%	46.8%	-	-	-	-
	Ours	**26.0%**	**56.7%**	**3.0**	**26.7%**	**56.5%**	**3.0**

Table 1. Video/Sentence retrieval on ActivityNet [17], LSMDC [32], and MSR-VTT [43]. Shown is the recall at a certain number of retrievals and MdR=median rank. We gray out models that used large-scale pre-training on HowTo100M [25]. We mark with ‡ models that used appearance, scene, motion, face, audio, OCR, ASR features from 11 different models. † indicates models that used appearance and motion. (*) denotes models use 2-second clips instead of the default 1-second clips.

where α_1 and α_2 are two margin parameters. The loss has three parts. The first compares the representation of entire video i and considers the matching paragraph, and an unmatched paragraph obtained as the matching paragraph of another video i'. Note that this term only applies in the case of video/paragraph matching. The second term is at the level of clips and sentences. It considers matching clips and sentences (matching video and paragraph, both with index i, and same clip and sentence, both with sentence j) and unmatching sentences of the same paragraph (same i different $j' \neq j$). The third term is similar, only the unmatched sentences are taken from an unmatched paragraph (different $i' \neq i$ and all possible j'). Note, we pad with zeros the long video/paragraph to have an equal length as the shorter video/paragraph.

3.4. Test Time Similarity Computation

To evaluate our video retrieval capabilities, we compute a similarity score S between every clip/sentence or video/paragraph pair. Given a clip c_i, we use the trained matching operator M (see (8)), *i.e.*,

$$\text{S}(\boldsymbol{c_j^i}, \boldsymbol{s_{j'}^{i'}}) = \text{M}\left(c_{jj'}^{ii'}, s_{j'j}^{i'i}\right). \tag{10}$$

Dataset	Method	Text2Vid			Vid2Text		
		R@1	R@5	MdR	R@1	R@5	MdR
ActivityNet	LSTM-YT [37]	0.0%	4.0%	102.0	0.0%	7.0%	98.0
	No Context [38]	5.0%	14.0%	78.0	7.0%	18.0%	56.0
	Dense full [17]	14.0%	32.0%	34.0	18.0%	36.0%	32.0
	FSE [49]	18.2%	44.8%	7.0	16.7%	43.1%	7.3
	HSE [49]	44.4%	76.7%	2.0	44.2%	76.7%	2.0
	Ours	**58.8%**	**89.2%**	**1.0**	**58.2%**	**88.1%**	**1.0**
DiDeMo	S2VT [38]	11.9%	33.6%	13.0	13.2%	33.6%	15.0
	FSE [49]	13.9%	36.0%.	11.0	13.1%	33.9%	12.0
	HSE [49]	30.2%	60.5%	3.3	30.1%	59.2%	3.0
	Ours	**38.5%**	**72.7%**	**2.7**	**38.7%**	**100.0%**	**2.4**
YouCook2	HSE [49]	43.5%	80.7%	2.0	43.8%	83.4%	2.0
	Ours	**49.1%**	**87.2%**	**1.0**	**51.3%**	**90.0%**	**1.0**
MSR-VTT	HSE [49]	32.9%	64.0%	3.0	32.5%	63.7%	3.0
	Ours	**38.0%**	**72.5%**	**2.0**	**37.8%**	**72.0%**	**2.0**

Table 2. Video/Paragraph retrieval on ActivityNet [17], DiDeMo [13] , YouCook2 [51] and MSR-VTT [43]. Shown is the recall at a certain number of retrievals and MdR=median rank.

Note, for video/sentence matching, all clips are concatenated into one clip. In the case of video paragraph retrieval, given a paragraph $\boldsymbol{p_{i'}}$, we calculate the similarity score based on a dot product between their dependent representation as follows:

$$\text{S}(\boldsymbol{v_i}, \boldsymbol{p_{i'}}) = \exp(-\|v_{ii'} - p_{i'i}\|). \tag{11}$$

4. Experiments

We evaluate our video and text matching method on a variety of datasets containing video clips with corresponding descriptions. To demonstrate the versatility of our model, we perform a comprehensive set of experiments in a diverse set of domains, such as movies, personal collections, and commercials. In addition to the matching task, we also explore weakly supervised localization in video and even video-guided machine translation. **Datasets:** We employ five different datasets, each having a different distribution of videos and a different descriptive style. For example, ActivityNet focuses on actions ("The girl dances around the room while the camera captures her movements.") while YouCook descriptions capture the order of very specific actions ("combine lemon juice sumac garlic salt and oil in a bowl"). The datasets we employ are listed next. **ActivityNet Dense Caption [17]** is a human activities video dataset, where each video has multiple events. The events are separated into clips, and each clip has a corresponding description. There are 849 video hours with 100,000 total sentences, giving rise to 10,009 videos with matching paragraphs for the training set, and 4,917/4,885(val1/val2) for the validation set. We follow previous works and report our results on val1. Results on val2 are available in the supplementary. **Distinct Describable Moments (DiDeMo) [13]** is collected from diverse personal videos and include everyday indoor scenes, nature scenes, events etc. The dataset consists of over 10,000 unedited videos. Each video comes with several actions described in natural language. The data is split into 8,395 train-

ing videos, 1,065 validation videos and 1,004 test videos. We report our results on the test-set. **YouCook2 [51]** contains 2,000 long untrimmed videos capturing the preparation of 89 cooking recipes, where each distinct recipe has 22 videos on average. The videos are provided with temporal boundaries and description sentences in imperative English. The dataset is split into a train set of size 1,500 and a test set with 500 videos. **Vatex [39]** contains 41,250 videos with 825,000 multilingual (English and Chinese) captions and includes over 600 fine-grained human activities. **MSR-VTT [43]** contains a total of 10K videos, each having 20 text descriptions. This dataset's standard split comprises 6,576 videos for training, 497 videos for validating, and 2,990 videos for testing. **LSMDC [32]** contains 118,081 short videos extracted from 202 movies, each described by a text caption extracted from the movie or the audio description. The test set is composed of 1000 videos.

4.1. Video/Sentence Retrieval

In this task, we match a single sentence and a video without clips. In datasets with clips, we treat each video as a single clip. Similarly, we concatenate all the captioning sentences into a single caption. In Tab. 1, we compare our method the state-of-the-art baselines. The baselines include the method of Liu *et al.* [21], who suggested using many video-expert networks, such as speech, audio, motion, OCR, appearance, and face detection. Another recent baseline is the Multimodal Transformer (MMT) [10], which encodes all the expert networks' representations jointly with self-attention. Our method, which considers only the appearance and text embeddings and does not use external knowledge, is much more straightforward, and also outperforms all those methods that did not use conditioned embeddings.

The most recent work clipBERT [18] study different frame sampling strategies. The approach we propose outperforms clipBERT on both ActivityNet and MSR-VTT by 4%, despite clipBERT's use of motion features. Notably, our technique outperforms pre-trained versions except for MMT, which also relies on many video-experts networks.

We also compare the efficiency of our architecture in terms of number of parameters. In total, our model has only 40.2mil parameters, while clipBERT, a Transformer-based network, has 199.3mil parameters. Additionally, MMT that considers various experts networks has 133.3mil parameters.

4.2. Video/Paragraph Retrieval

Our next task is to match videos with descriptive paragraphs. Compared to the video/sentence retrieval, this variant uses clips segmentation and corresponding sentences. Each corresponding clip and sentence are conditioned separately. Note that we also employ global conditioning (*i.e.*, the global video and paragraph embeddings). The second hierarchy is used to handle the conditioned clips and sentences representations (see Sec. 3.2). Tab. 2 reports the results

Model	Text2Vid		Vid2Text	
	R@1	R@5	R@1	R@5
architecture				
w/o 2nd H	33.0%	66.7%	32.2%	65.8%
w/o M match	57.2%	87.9%	56.9%	87.1%
w/o global	57.2%	87.9%	57.0%	87.4%
conditioning				
w/o attn	49.1%	80.6%	47.4%	80.2%
w/o frames attn	52.7%	82.0%	51.3%	82.3%
w/o sentence attn	54.6%	84.5%	52.9%	83.4%
w/o global attn	56.6%	87.3%	56.2%	86.9%
alternative attn	56.1%	87.3%	55.6%	86.8%
loss				
w/o clip match	52.9%	84.1%	52.3%	82.2%
embedding-dimension				
$d_h = 256$	54.9%	87.1%	53.9%	85.7%
$d_h = 1024$	**58.9%**	**90.1%**	57.8%	87.8%
Full	58.8%	89.2%	**58.2%**	**88.1%**

Table 3. An ablations study on ActivityNet.

of our method in comparison to the results collected from the literature as well as HSE results we ran, ourselves, on YouCook2 , and MSRT-VTT. HSE [49] also uses hierarchy encoding. Our loss is considerably simpler than that of HSE. Moreover, our approach leverage conditioned embeddings. Our method obtains an improvement in R@1 over HSE result in both matching directions of 14% in ActivityNet, 9.6% in DiDeMo, 7.5% in YouCook2 and 5.3% in MSR-VTT.

4.2.1 Ablation study

In Tab. 3 we assess several design choices evaluated on video and paragraph retrieval setup: 1) We examine the importance of different architecture components. In 'w/o 2nd H', we used only one hierarchy by treating the output of our global embedding as the conditioned sentence and video representations (*i.e.*, $v_{ii'}^0, p_{ii'}^0$). While the performance dropped significantly, this score is impressive because the model does not take into account the events' temporal locations (*i.e.*, clips). In 'w/o M match', we evaluate our model with a normalized dot-product instead of our trainable matching module M. Last, in 'w/o-global', we do not use a global video and paragraph embeddings to initialize GRU_v and GRU_p. In this case, the performance drops by 1% on R@1. Overall, we show that all the integral components in our architecture contribute to the performance. 2) We assess our conditioning approach. We treat it as a form of co-attention, *i.e.*, picking the relevant frames and words. In 'w/o-atten', we show the importance of attention, omitting the attention leads to a significant drop in performance (47.4% *vs.* 58.2%). In 'w/o text attn', we infer the clip's attention as usual, but instead of text attention, we performed a max-pooling across the sentence representation. As a result, the performance drops by 5.3% on R@1. Similarly, in 'w/o frame attn', we

Low-relevance frames High-relevance frames Low-relevance frames High-relevance frames
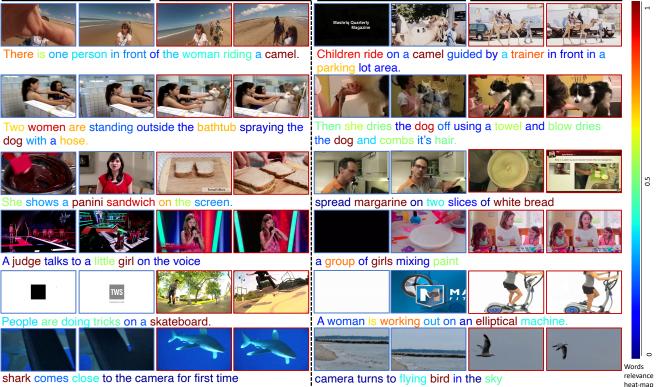
There is one person in front of the woman riding a camel.

Children ride on a camel guided by a trainer in front in a parking lot area.

Two women are standing outside the bathtub spraying the dog with a hose.

Then she dries the dog off using a towel and blow dries the dog and combs it's hair.

She shows a panini sandwich on the screen.

spread margarine on two slices of white bread

A judge talks to a little girl on the voice

a group of girls mixing paint

People are doing tricks on a skateboard.

A woman is working out on an elliptical machine.

shark comes close to the camera for first time

camera turns to flying bird in the sky

Words relevance heat-map

Figure 4. Our approach also serves the additional benefit of describing the specific frames the model uses for retrieval. We illustrate queries, relevancy of frames μ_c and words μ_s derived from Eq. (4). We show results on ActivityNet (1st, 2nd, 5th rows), YouCook2 (3rd row), DiDeMo (6th row), MSR-VTT (4th row). Each row displays two queries and a clip from the retrieved video. In the clip, we show two of the most relevant frames to the query (marked with red borders) and two of the least relevant frames (marked with blue borders). We also illustrated relevant words to the clip using color maps.

removed the clips' attention, which leads to a significant performance drop of 7% on R@1. In 'w/o global attn', we use max polling for the global vector instead of attention, which leads to a drop of 2% on R@1. In 'alternative attn', we replace our attention with that of [48], which leads to a 2.6% drop on R@1. Additionally, our version runs 36% faster. 3) Removing the auxiliary clip and sentence matching drops the performance by 7.2% on R@1. 4) We change our main embedding parameter d_h. It is evident that our method is stable to this parameter in terms of Recall@k metric and it outperforms state of the art (Tab. 2) for multiple values.

4.2.2 Qualitative analysis

In Fig. 4 we use the interaction scores (see Eq. (4)) to identify the relevant frames and words. It simplifies explanations since finding the frames responsible for retrieving a specific long video can be challenging. In the first step, we highlight the words' relevance through a heatmap. We find that most of the relevant words are grounded nouns. For example, in the first row, the relevant words in the query are 'children,' 'camel,' 'trainer,' 'parking.' These are useful words for visual matching. Next, we randomly selected a clip from the retrieved video and displayed the two frames that got

the highest interaction scores (as indicated by the red border). The bottom two frames, are marked by blue borders. The bottom frames are usually irrelevant to the query. For instance, in the 2nd row, on the left side, the query is "... spraying the dog," but the bottom frames do not depict any dog. We also find the frames to be uninformative. In the first row, on the left, the woman's hand hides most of the details. In addition, clips often have frames with production logos, *e.g.*, the bottom frames in 5th-row videos. We also show a complicated case of two similar queries. Both queries in the first row deal with camels; in the second row, both queries deal with dogs; in the third row, both videos deal with bread sandwiches. Thus, our model identified subtle details to retrieve the correct video. The 4th row shows samples from the MSR-VTT dataset. The top 2 frames in the left examples are related to 'girl' as highlighted in the query attention, and for the right example, we can see that the top 2 frames contain 'group of girls.' In the 3rd row, we show two queries from Youcook2 and find our approach to pick the relevant frames showing the query's ingredients (*e.g.*, a sandwich, margarine, white bread). Lastly, in the 6th row, we show queries from DiDeMo. The attention behaves well by selecting relevant frames, *e.g.*, a shark, and a bird.
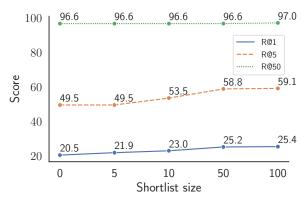
Figure 5. Comparison of different shortlist sizes. A larger shortlist increases performance at the expense of latency.

N	Alg.	Text2Vid		Vid2Text	
		R@1	R@5	R@1	R@5
1	HSE	18.0%	45.5%	16.5%	44.9%
	Ours	**21.9%**	**49.3%**	**21.5%**	**49.1%**
2	HSE	20.0%	48.9%	18.4%	47.6%
	Ours	**24.8%**	**55.6%**	**24.3%**	**54.6%**
3	HSE	20.0%	48.6%	18.2%	47.9%
	Ours	**24.9%**	**55.8%**	**24.7%**	**55.1%**
4	HSE	20.5%	49.3%	18.7%	48.1%
	Ours	**24.9%**	**55.9%**	**24.8%**	**55.3%**

Table 4. Video/Paragraph retrieval using N fixed-length segments on the text-truncated ActivityNet.

Method	en→zh	zh→en
Vatex [39]	29.12%	26.42%
Ours	**32.34%**	**28.11%**

Table 5. Video-guided translation between English (en) and Chinese (zh). BLEU-4 scores.

4.3. Inference time and shortlist

For each inference call, our approach requires recomputing the representations. This computational burden can be alleviated through the use of a static method (*e.g.*, HSE [49]) to retrieve a shortlist of relevant candidates. In Fig. 5, we examine different shortlist sizes and find that a shortlist of 50 already allows most of the performance gain. We measured the inference time on ActivityNet in the paragraph/video task. The inference latency for a single inference call is ~0.4s for a shortlist of 5, ~1.1s for a shortlist of 10, ~1.8s for a shortlist of 50, and ~3.2s for a shortlist of 100. The overall inference time of ActivityNet test set for a shortlist of 50 is ~7 minutes, in comparison to ~50 minutes without a shortlist and ~50s using the HSE method as static approach. Note, inference of advanced static approaches also takes a longer time compared to HSE. For example, MSE [10] inference takes about four minutes over the test set.

4.4. Weakly Supervised Retrieval

Many scenarios do not allow supervision of the events' temporal location. Multiple-sentence descriptions are also expensive. Weakly supervised retrieval is similar to the previous retrieval task, but excludes the events segmentation and adds a fixed number of sentences. In Tab. 4, following [49], we report results obtained when breaking the paragraph into sentences. We replace the ground-truth events in the ActivityNet dataset with equal-length non-overlapping temporal segments. For the textual input, in case there are fewer segments than sentences, we truncate the paragraph. If there are more segments than sentences, we repeat the last sentence. The absence of ground-truth events and truncating the paragraph harms the performance (24% drop on R@1). Still, our model outperforms the previous baseline by 5.0-6.5% on R@1 depending on the number of segments.

4.5. Video-guided Machine Translation

Finally, we extend the use of our conditioned embeddings to other tasks. We consider the task of Video-guided Machine Translation (VMT) [39], in which a sentence in

a source language is translated to a target language, using both the source-text and a matching video. Our solution starts with training our retrieval model to embed the video and text closer together. Since the videos are not segmented into clips, our model only utilizes the first hierarchy. We then fine-tune the video and sentence representations for the VMT, by using an LSTM decoder, followed by MLE loss on words. Like [39], our LSTM has 1024 cells and a beam search with a width of 5 was used for evaluation. As is evident from the results, shown in Tab. 5, our method outperforms the method of [39].

5. Conclusions

For retrieval, it is convenient to embed the data in each domain statically, which can be pre-processed and stored for future use. Nevertheless, our method highlights the importance of interactions or the cross-domain context when embedding the data of the two domains that are being matched. Meanwhile, if complexity is an issue for a specific application, one can consider using a shortlist that is obtained using static embeddings. In the future, we believe that combining conditioned embeddings and large-scale pre-training can lead to a significant performance gain.

Acknowledgments

References

[1] Hrishikesh Aradhye, George Toderici, and Jay Yagnik. Video2text: Learning to annotate video content. In *ICDM Workshops*, 2009.

[2] Hedi Ben-Younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. Mutan: Multimodal tucker fusion for visual question answering. In *ICCV*, 2017.

[3] Tom Braude, Idan Schwartz, Alexander Schwing, and Ariel Shamir. Towards coherent visual storytelling with ordered image attention. *arXiv preprint arXiv:2108.02180*, 2021.

[4] Shaoxiang Chen and Yu-Gang Jiang. Semantic proposal for activity localization in videos via sentence query. In *AAAI*, 2019.

[5] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. In *CVPR*, 2020.

[6] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *SSST-8*, 2014.

[7] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. Dual encoding for zero-example video retrieval. In *CVPR*, 2019.

[8] Xuguang Duan, Wenbing Huang, Chuang Gan, Jingdong Wang, Wenwu Zhu, and Junzhou Huang. Weakly supervised dense event captioning in videos. In *NIPS*, 2018.

[9] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *EMNLP*, 2016.

[10] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. *ECCV*, 2020.

[11] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *ICCV*, pages 5267–5275, 2017.

[12] Mingfei Gao, Larry S Davis, Richard Socher, and Caiming Xiong. Wslln: Weakly supervised natural language localization networks. *EMNLP*, 2019.

[13] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *ICCV*, 2017.

[14] Haiqi Huang, Yueming Lu, Fangwei Zhang, and Songlin Sun. A multi-modal clustering method for web videos. In *ISCTCS*, 2012.

[15] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. In *NIPS*, 2018.

[16] Jin-Hwa Kim, Kyoung-Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Hadamard product for low-rank bilinear pooling. *ICLR*, 2017.

[17] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, 2017.

[18] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. *CVPR*, 2021.

[19] Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. A hierarchical neural autoencoder for paragraphs and documents. *ACL*, 2015.

[20] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. *EMNLP*, 2020.

[21] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. Use what you have: Video retrieval using representations from collaborative experts. *BMVC*, 2019.

[22] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. In *NIPS*, 2016.

[23] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*, 2020.

[24] Bar Mayo, Tamir Hazan, and Ayellet Tal. Visual navigation with spatial attention. In *CVPR*, 2021.

[25] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019.

[26] Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metze, and Amit K Roy-Chowdhury. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In *ICMR*, 2018.

[27] Niluthpol Chowdhury Mithun, Sujoy Paul, and Amit K Roy-Chowdhury. Weakly supervised video moment retrieval from text queries. In *CVPR*, 2019.

[28] Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. Hierarchical multimodal lstm for dense visual-semantic embedding. In *ICCV*, 2017.

[29] Pingbo Pan, Zhongwen Xu, Yi Yang, Fei Wu, and Yueting Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. In *CVPR*, 2016.

[30] Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander Hauptmann, João Henriques, and Andrea Vedaldi. Support-set bottlenecks for video-text representation learning. *ICLR*, 2020.

[31] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[32] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A dataset for movie description. In *CVPR*, 2015.

[33] Idan Schwartz, Alexander G Schwing, and Tamir Hazan. High-order attention models for visual question answering. *NIPS*, 2017.

[34] Idan Schwartz, Alexander G Schwing, and Tamir Hazan. A simple baseline for audio-visual scene-aware dialog. In *CVPR*, 2019.

[35] Idan Schwartz, Seunghak Yu, Tamir Hazan, and Alexander G Schwing. Factor graph attention. In *CVPR*, 2019.

[36] Reuben Tan, Huijuan Xu, Kate Saenko, and Bryan A. Plummer. Logan: Latent graph co-attention network for weakly-supervised video moment retrieval. *WACV*, 2021.

[37] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *ICCV*, 2015.

[38] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. *ACL*, 2015.

[39] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *ICCV*, 2019.

[40] Shikui Wei, Yao Zhao, Zhenfeng Zhu, and Nan Liu. Multi-modal fusion for video search reranking. *TKDE*, 2009.

[41] Huijuan Xu, Kun He, Bryan A Plummer, Leonid Sigal, Stan Sclaroff, and Kate Saenko. Multilevel language and vision integration for text-to-clip retrieval. In *AAAI*, 2019.

[42] Huijuan Xu and Kate Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *ECCV*, 2016.

[43] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016.

[44] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

[45] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *CVPR*, 2016.

[46] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *CVPR*, 2016.

[47] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *ECCV*, 2018.

[48] Yitian Yuan, Tao Mei, and Wenwu Zhu. To find where you talk: Temporal sentence localization in video with attention based location regression. In *AAAI*, 2019.

[49] Bowen Zhang, Hexiang Hu, and Fei Sha. Cross-modal and hierarchical modeling of video and text. In *ECCV*, 2018.

[50] Da Zhang, Xiyang Dai, Xin Wang, Yuan-Fang Wang, and Larry S Davis. Man: Moment alignment network for natural language moment retrieval via iterative graph adjustment. In *CVPR*, 2019.

[51] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018.

[52] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *CVPR*, 2020.