A Hierarchical Multi-Modal Encoder for Moment Localization in Video Corpus

Bowen Zhang[1*†] zhan734@usc.edu
Hexiang Hu[1*†] hexiangh@usc.edu
Joonseok Lee[2†] joonseok@google.com
Ming Zhao[2] astroming@google.com
Sheide Chammas[2] sheide@google.com
Vihan Jain[2] vihanjain@google.com
Eugene Ie[2] eugeneie@google.com
Fei Sha[2‡] fsha@google.com

[1]U. of Southern California [2]Google Research

Abstract

Identifying a short segment in a long video that semantically matches a text query is a challenging task that has important application potentials in language-based video search, browsing, and navigation. Typical retrieval systems respond to a query with either a whole video or a pre-defined video segment, but it is challenging to localize undefined segments in untrimmed and unsegmented videos where exhaustively searching over all possible segments is intractable. The outstanding challenge is that the representation of a video must account for different levels of granularity in the temporal domain. To tackle this problem, we propose the HierArchical Multi-Modal EncodeR (HAMMER) that encodes a video at both the coarse-grained clip level and the fine-grained frame level to extract information at different scales based on multiple subtasks, namely, video retrieval, segment temporal localization, and masked language modeling. We conduct extensive experiments to evaluate our model on moment localization in video corpus on ActivityNet Captions and TVR datasets. Our approach outperforms the previous methods as well as strong baselines, establishing new state-of-the-art for this task.

1. Introduction

With over 70% of the current internet traffics being video data [26], a growing number of videos are being created, shared, and consumed over time. To effectively and efficiently search, browse, and navigate video contents, an intelligent system needs to understand the rich and complex semantic information in them. For this type of use cases, the recently proposed task of *moment localization in video corpus* (MLVC) highlights several challenges in semantic understanding of videos [8, 20]. The goal of MLVC is to find a video segment that corresponds to a text query from a corpus of *untrimmed and unsegmented* videos, with a significant amount of variation in factors such as the type of contents, lengths, visual appearance, quality, and so on.

This task can be seen as "finding a needle in the haystack". It is different from searching videos with broad queries such as genres or names of the artists. In contrast, the text query needs to be semantically congruent to a relatively short segment in a much longer target video. For example, the query "LeBron James shot over Yao Ming" matches only a few seconds of clips in a game of hours long. Thus, MLVC requires semantic understanding of videos at a more fine-grained level than video retrieval, which typically only targets the whole video. Furthermore, finding the corresponding segment for a text query requires combing through all videos in a corpus and all possible segments in each video. For a large corpus with long videos, it is not feasible to have such computational complexity that depends on the square of the (averaged) number of frames.

In this paper, we address this challenge by representing videos at multiple scales of granularity. At the coarse-grained level, the representation captures semantic information in a video over long temporal spans (*e.g.*, clips), allowing us to retrieve the most relevant set of videos for a text query. At the fine-grained level, the representation captures semantic information in short temporal spans (*e.g.*, frames) to allow for precise localization of the most relevant video segments among the retrieved videos.

We propose a novel hierarchical multi-modal encoder (HAMMER) to implement this idea. HAMMER uses cross-modal attention to combine the information between the text and visual modalities. The cross-modal learning occurs hierarchically at 3 scales: frame, clip, and video (as a whole). Frames are the most fine-grained building blocks of a video. Each clip consists of a non-overlapping set of frames with equal length, and is in turn the building block of the final video-level representation. The architecture of the model

[*] Work done while at Google
[†] Authors Contributed Equally
[‡] On leave from USC (feisha@usc.edu)

is illustrated in Fig. 1. The frame-level representation is obtained from a text-visual cross-modal encoder operated on video frames, while the clip-level representation is built upon the frame-level representation with a similar encoder.

The introduction of clip-level representation encoder is important as it allows us to capture both coarse- and fine-grained semantic information. In contrast, existing approaches for MLVC [8, 20] and other visual-language tasks [3, 9, 11, 18] typically pack information of different granularity into a single vector embedding, making it hard to balance the differing demands between retrieving a long video and localizing a short segment.

We apply HAMMER to MLVC task on two large-scale datasets, ActivityNet Captions [19] and TVR [20]. We train it with a multi-tasking approach combining three objectives: video retrieval, temporal localization, and an auxiliary masked language modeling. Our experiments demonstrate the efficacy of HAMMER and establish state-of-the-art performance on all the tasks simultaneously—video retrieval, moment localization in single video and moment localization in video corpus. To better understand the inner-workings of our model, we compare it with a strong FLAT baseline, a video encoder without any hierarchical representation. Since the longer videos tend to be less homogeneous, it becomes decidedly important to represent the videos at multiple levels of granularity. Our analysis shows that the performance of a FLAT baseline declines, when the number of frames irrelevant to the text query increases. On the other hand, the performance of our proposed HAMMER model is robust to the length of the videos, showing that our hierarchical approach is not affected by the increase of irrelevant information and can flexibly handle longer videos.

Our contributions are summarized as follows:

- We propose a novel model architecture HAMMER that represents videos hierarchically and improves video modeling at long-term temporal scales.
- We demonstrate the efficacy of HAMMER on two large-scale datasets, *i.e.*, ActivityNet Captions and TVR, outperforming previous state-of-the-art methods.
- We carry out a detailed analysis of HAMMER and show that it particularly improves the performance of video retrieval over strong baselines on long videos.
- We conduct a thorough ablation study to understand the effects of different design choices in our model.

2. Related Work

Most existing MLVC approaches consider text-based video retrieval [7, 25, 27, 32, 36] and temporal localization [2, 12, 14, 22, 28, 35] as separate tasks.

Video Retrieval (VR) is a task that ranks candidate videos based on their relevance to a descriptive text query. Standard cross-modal retrieval methods [19, 33] represent both video and text as two holistic embeddings, which are then used for computing the similarity as the ranking score. When the text query is a lengthy paragraph, hierarchical modeling is applied to both modalities separately [29, 39], leading to a significant improvement on the performance of text-based video retrieval. Different from prior work, in this study we consider a more realistic problem where we use a single query sentence that describes only a small segment to retrieve the entire video. For instance, the text query "*Add the onion paste to the mixture*" may corresponds to a temporal segment of a few seconds in a long cooking video.

Temporal Localization (TL) aims at localizing a video segment (usually a short fraction of the entire video) described by a query sentence inside a video. Two types of methods have been proposed to tackle this challenge, namely the top-down (or proposal-based) approach [12, 14, 35] and the bottom-up (or proposal-free) approach [2, 4, 23, 37, 38]. The top-down approach first generates multiple clip proposals before matching them with a query sentence to localize the most relevant clip from the proposals. The bottom-up approach first calculates a query-aware video representation as a sequence of features, then predicts the start and end times of the clip described by the query.

Moment Localization in Video Corpus (MLVC) is first proposed by Escorcia *et al.* [8]. They consider a practical setting where they require models to jointly retrieve videos and localize moments corresponding to natural language queries from a large collection of untrimmed and unsegmented videos. They devised a top-down localization approach that compares text embeddings on uniformly partitioned video chunks. Recently, Lei *et al.* [20] proposed a new dataset, TVR, that considers a similar task called Moment Retrieval in Video-Subtitle Corpus, which requires a model to align a query text temporally with a video clip, using multi-modal information from video and *subtitles* (derived from Automatic Speech Recognition or ASR).

3. Method

We first describe the problem setting of MLVC and introducing the notations in §3.1. In §3.2, we describe a general strategy of decomposing MLVC into two sub-tasks, VR and TL [23, 34]. The main purpose is to reduce computation and to avoid the need to search all possible segments of all videos. In §3.3, we present a novel HierArchical Multi-Modal EncodeR (HAMMER) model and describe how it is trained in §3.4. Finally, we describe key details for inference in §3.5.

3.1. Problem Setting and Notations

We represent a video $\boldsymbol{v}$ as a sequence of N frames $\{\boldsymbol{x}_t | t = 1, \dots, \mathsf{N}\}$, where $\boldsymbol{x}_t$ is a visual feature vector representing the t-th frame. Given a text query $\boldsymbol{h}$ (*e.g.*, a sen-

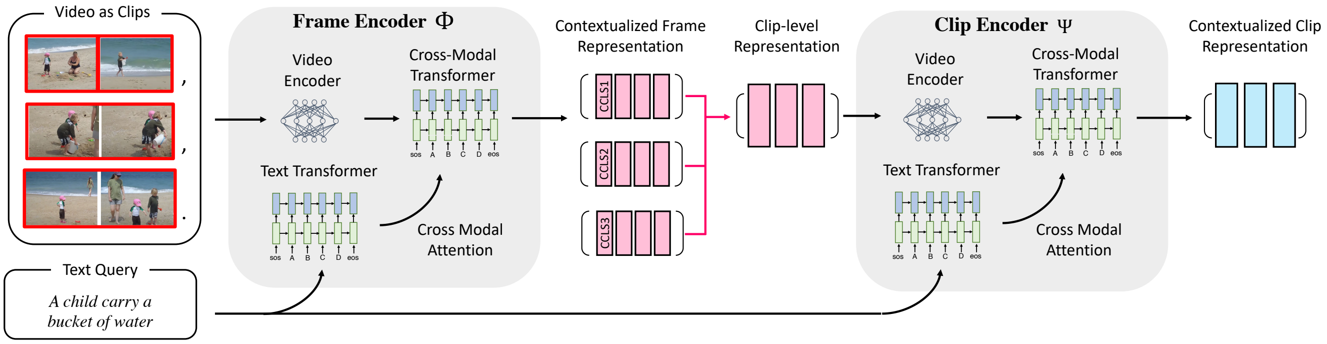

Figure 1. Overview of the HAMMER model. The model contains two cross-modal encoders, a frame encoder and a clip encoder on top of it. The outputs of the model are contextualized frame-level and clip-level features, which are used by downstream task-specific modules, *e.g.* video retrieval and temporal localization.

tence), our goal is to learn a parameterized function (*i.e.*, neural networks) that accurately estimates the conditional probability $p(\boldsymbol{s}|\boldsymbol{h})$, where $\boldsymbol{s}$ is a video segment given by $\boldsymbol{s} = \{\boldsymbol{x}_t | t = t^{\text{s}}, \ldots, t^{\text{e}}\}$. t^{s} and t^{e} stand for the indices of the starting and the ending frames of the segment in a video $\boldsymbol{v}$. Note that for a video corpus $\mathcal{V}$ with an average length of N frames, the number of all possible segments is $O(|\mathcal{V}| \times \mathsf{N}^2)$. Thus, in a large corpus, exhaustive search for the best segment $\boldsymbol{s}$ corresponding to $\boldsymbol{h}$ is not feasible. In what follows, we describe how to address this challenge.

To localize the moment $\boldsymbol{s}$ that best corresponds to a text query $\boldsymbol{h}$, we need to identify

$$\boldsymbol{s}^* = \underset{\boldsymbol{s}}{\operatorname{argmax}}\, p(\boldsymbol{s}|\boldsymbol{h}) = \underset{\boldsymbol{s}}{\operatorname{argmax}} \sum_{\boldsymbol{v}} p(\boldsymbol{s}|\boldsymbol{v}, \boldsymbol{h}) p(\boldsymbol{v}|\boldsymbol{h}) \quad (1)$$

Note that the conditional probability is factorized into two components. If we assume $\boldsymbol{s}$ uniquely belongs to only one video in the corpus $\mathcal{V}$, then the marginalization over the video $\boldsymbol{v}$ is vacuous and can be discarded. This leads to

$$\max_{\boldsymbol{s}} p(\boldsymbol{s}|\boldsymbol{h}) = \max_{\boldsymbol{v}} \max_{\boldsymbol{s} \in \boldsymbol{v}} p(\boldsymbol{s}|\boldsymbol{v}, \boldsymbol{h}) p(\boldsymbol{v}|\boldsymbol{h}). \quad (2)$$

The training data are available in the form of $(\boldsymbol{h}^{(i)}, \boldsymbol{v}^{(i)}, \boldsymbol{s}^{(i)})$ where $\boldsymbol{s}^{(i)} \subset \boldsymbol{v}^{(i)}$ is the matched segment to the query $\boldsymbol{h}^{(i)}$.

3.2. Two-Stage MLVC: Retrieval and Localization

As aforementioned, this inference of Eq. (2) is infeasible for large-scale corpora and/or long videos. Thus, we approximate it by

$$\boldsymbol{s}^* \approx \underset{\boldsymbol{s} \in \boldsymbol{v}^*}{\operatorname{argmax}}\, p(\boldsymbol{s}|\boldsymbol{v}^*, \boldsymbol{h}), \text{ with } \boldsymbol{v}^* = \underset{\boldsymbol{v}}{\operatorname{argmax}}\, p(\boldsymbol{v}|\boldsymbol{h}) \quad (3)$$

This approximation allows us to build two different learning components and stage them together to solve MLVC. This approach has been applied in a recent work on the task [20]. We give a formal summary below.

Video Retrieval (VR) identifies the best video $\boldsymbol{v}^*$ by minimizing the negative log-likelihood of $p(\boldsymbol{v}|\boldsymbol{h})$

$$\ell^{\text{VR}} = -\sum_i \log p(\boldsymbol{v}^{(i)}|\boldsymbol{h}^{(i)}) \quad (4)$$

where $\boldsymbol{v}^{(i)}$ is the ground-truth video for the text query $\boldsymbol{h}^{(i)}$. This is a rather standard (cross-modal) retrieval problem, which has been widely studied in the literature. (See §2 for some references.)

Temporal Localization (TL) models $p(\boldsymbol{s}|\boldsymbol{v}, \boldsymbol{h})$. While it is possible to model $O(\mathsf{N}^2)$ possible segments in a video with N frames, we choose to model it with the probabilities of identifying the correct starting (t^{s}) and ending (t^{e}) frames:

$$p(\boldsymbol{s}|\boldsymbol{v}, \boldsymbol{h}) \approx p(t^{\text{s}}|\boldsymbol{v}, \boldsymbol{h}) \cdot p(t^{\text{e}}|\boldsymbol{v}, \boldsymbol{h}) \cdot \mathbb{I}[t^{\text{e}} > t^{\text{s}}] \quad (5)$$

Here, we consider t^{s} and t^{e} to be independent to efficiently approximate $p(\boldsymbol{s}|\boldsymbol{v}, \boldsymbol{h})$. The indicator function $\mathbb{I}[\cdot]$ simply stipulates that the ending frame needs to be after the starting frame.

To model each of the factors, we treat it as a frame classification problem, annotating each frame with one of the three possible labels: BEGIN and END marks the starting and ending frames respectively, with all other frames as OTHER. We denote this as B, E, O classification scheme. During training, we optimize (the sum of) the frame-wise cross-entropy between the model's predictions and the labels. We denote the training loss as

$$\ell^{\text{TL}} = -\sum_i \sum_t f_t^{(i)} \log p(y_t^{(i)}|\boldsymbol{v}^{(i)}, \boldsymbol{h}^{(i)}), \quad (6)$$

where $f_t^{(i)}$ is the true label for the frame $\boldsymbol{x}_t$ of the video $\boldsymbol{v}^{(i)}$, and $y_t^{(i)}$ is the corresponding prediction of the model.

This type of labeling schemes have been widely used in the NLP community, for example, recently for span-based question and answering [10, 16].

3.3. HierArchical Multi-Modal Encoder (HAMMER)

Our first contribution is to introduce the hierarchical modeling approach to parameterize the conditional probability $p(\boldsymbol{v}|\boldsymbol{h})$ for the VR sub-task and the labeling model $p(y|\boldsymbol{v},\boldsymbol{h})$ for the TL sub-task. In the next section, we describe novel learning algorithms for training our model.

Main idea Video and text are complex and structural objects. They are naturally in "temporally" linear orders of frames and words. More importantly, semantic relatedness manifests in both short-range and long-range contextual dependencies. To this end, HAMMER infuses textual and visual information hierarchically at different temporal scales. Figure 1 illustrates the architecture of HAMMER. A key element here is to introduce cross-modal attention at both the frame level and the clip level.

Clip-level Representation We introduce an intermediate-level temporal unit with a fixed length of M frames, and refer to them as a clip $\boldsymbol{c}_k = \{\boldsymbol{x}_t | t = (k-1)\cdot\mathsf{M}, \ldots, k\cdot\mathsf{M}-1\}$, where $k = 1,\ldots,\lceil\mathsf{N}/\mathsf{M}\rceil$. As such, a video can also be hierarchically organized as a sequence of non-overlapping video clips $\boldsymbol{v} = \{\boldsymbol{c}_k | k = 1,\ldots,\lceil\mathsf{N}/\mathsf{M}\rceil\}$. M is a hyper-parameter to be adjusted on different tasks and datasets. We emphasize while sometimes segments and clips are used interchangeably, we refer to "segment" as a set of frames that are also the visual grounding of a text query, and "clip" as a collection of temporally contiguous frames. We treat them as holding memory slots for aggregated lower-level semantic information in frames.

Cross-modal Transformers HAMMER has two cross-modal Transformers. At the frame-level, the frame encoder Φ takes as input both the frame sequence of a video clip and the text sequence of a query, and outputs the contextualized visual frame features $\{\Phi(\boldsymbol{x}_t;\boldsymbol{c}_k,\boldsymbol{h})\}$ for each clip $\boldsymbol{c}_k$. The frame encoder Φ encodes the local and short-range contextual dependencies among the frames of the same clip.

We also introduce a *Clip CLS Token* ($\texttt{CCLS}_k$) for each $\boldsymbol{c}_k$ [24]. The contextual embedding of this token gives the representation of the clip:

$$\boldsymbol{\phi}_k = \Phi(\texttt{CCLS}_k;\boldsymbol{c}_k,\boldsymbol{h}) \tag{7}$$

Contextual embeddings for all clips are then fed into a higher-level clip encoder Ψ, also with cross-modal attention to the input text, yielding a set of contextualized clip representation

$$\Psi_{\boldsymbol{v}} = \{\Psi(\phi_k;\boldsymbol{v},\boldsymbol{h}) \mid k = 1,\ldots,\lceil\mathsf{N}/\mathsf{M}\rceil\}. \tag{8}$$

Note that $\Psi_{\boldsymbol{v}}$ now encodes the global and longer-range contextual dependencies among all frames (through clips)[1].

To summarize, our model has 3 levels of representations: the contextualized frames $\{\Phi(\boldsymbol{x}_t;\boldsymbol{c}_k,\boldsymbol{h})\}$, the clips $\{\boldsymbol{\phi}_k\}$, and the entire video $\Psi_{\boldsymbol{v}}$. Next, we describe how to use them to form our learning algorithms.

3.4. Learning HAMMER for MLVC

The different levels of representation allows for the flexibility for modeling the two subtasks (VR and TL) with semantic information across different temporal scales.

Modeling Video Retrieval We use the contextualized clips to compute the video-query compatibility score for a query $\boldsymbol{h}$ and its corresponding video $\boldsymbol{v}$. In order to retrieve the likely relevant videos as much as possible, we need a coarse-grained matching that focuses more on higher-level semantic information.

Specifically, we identify the best matching among all clip embeddings $\Psi(\boldsymbol{c}_k;\boldsymbol{v},\boldsymbol{h})$ and use it as the matching score for the whole video:

$$p(\boldsymbol{v}|\boldsymbol{h}) \propto f(\boldsymbol{v},\boldsymbol{h}) = \max_k \left(\{\boldsymbol{\theta}_{\text{VR}}^\top \cdot \Psi(\boldsymbol{\phi}_k;\boldsymbol{v},\boldsymbol{h})\}\right) \tag{9}$$

where $\boldsymbol{\theta}_{\text{VR}}$ is a linear projection to extract the matching scores[2]. The conditional probability is normalized with respect to all videos in the corpus (though in practice, a set of positive and negative ones).

Modeling Temporal Localization As in the previous section, we treat localization as classifying a frame into $\texttt{B}$, $\texttt{E}$, or $\texttt{O}$:

$$p(y_t|\boldsymbol{v},\boldsymbol{h}) \approx p(\boldsymbol{c}_k|\boldsymbol{v},\boldsymbol{h})\cdot p(y_t|\boldsymbol{c}_k,\boldsymbol{h}) \tag{10}$$

Note that each frame can belong to only one clip $\boldsymbol{c}_k$ so there is no need to marginalize over $\boldsymbol{c}_k$. The probability $p(\boldsymbol{c}_k|\boldsymbol{v},\boldsymbol{h})$ measures the likelihood of $\boldsymbol{c}_k$ containing a label y_t in one of its frames. The second factor measures the likelihood that the specific frame $\boldsymbol{x}_k$ is labeled as y_t. Clearly, these two factors are on different semantic scales and are thus modeled separately:

$$p(\boldsymbol{c}_k|\boldsymbol{v},\boldsymbol{h}) \propto \boldsymbol{u}^\top \cdot \ [\Psi(\phi_k;\boldsymbol{v},\boldsymbol{h}),\ \Psi(\texttt{TCLS};\boldsymbol{v},\boldsymbol{h})\ ] \tag{11}$$

$$p(y_t|\boldsymbol{c}_k,\boldsymbol{h}) \propto \boldsymbol{w}_{y_t}^\top \cdot [\Phi(\boldsymbol{x}_t;\boldsymbol{c}_k,\boldsymbol{h}),\Phi(\texttt{TCLS};\boldsymbol{c}_k,\boldsymbol{h})] \tag{12}$$

where $\texttt{TCLS}$ is a text $\texttt{CLS}$ token summarizing the query embedding.

Masked Multi-Modal Model Masked language modeling has been widely adopted as a pre-training task for language modeling [6, 24, 30]. The main idea is to backfill a masked text token from its contexts, *i.e.*, the other tokens in a sentence.

The multi-modal modeling task in this paper can similarly benefit from this idea. During training, we mask randomly some text tokens. We expect the model to achieve two things:

[1] Alternatively, we can summarize it (into a vector, in lieu of the set) through various reduction operations such as pooling or introducing a video-level *CLS token* $\texttt{VCLS}$.

[2] An alternative design is to pool all $\Psi(\boldsymbol{\phi}_k;\boldsymbol{v},\boldsymbol{h})$ and then perform a linear projection. However, this type of polling has a disadvantage that a short but relevant segment – say within a clip – can be overwhelmed by all other clips. Empirically, we also find the current formalism works better. A similar finding is also discovered in [39].

(1) using the partially masked text query to retrieve and localize which acts as a regularization mechanism; (2) better text grounding by recovering the masked tokens with the assistance of the multimodal context, i.e., both the textual context and the visual information in the frames and the clips.

To incorporate a masked query to the loss functions ℓ^{VR} and ℓ^{TL} of the model we apply $\boldsymbol{h} \otimes (\boldsymbol{1} - \boldsymbol{m})$ to replace $\boldsymbol{h}$, where $\boldsymbol{m}$ is a binary mask vector for text tokens, $\boldsymbol{1}$ is a one-valued vector of the same size, and $\otimes$ indicates element-wise multiplication. We introduce another loss to backfill the missing tokens represented by $\boldsymbol{h} \otimes \boldsymbol{m}$:

$$\ell^{\text{MASK}} = -\log p(\boldsymbol{h} \otimes \boldsymbol{m} | \boldsymbol{v}, \boldsymbol{h} \otimes (\boldsymbol{1} - \boldsymbol{m})) \qquad (13)$$

This probability is computed using both $\Phi(\cdot)$ for frames and $\Psi(\cdot)$ for clips.

Multi-Task Learning Objective We use a weighted combination of video retrieval, moment localization, and masked multi-modal modeling objectives as our final training objective:

$$\ell = \mathbb{E}_{\boldsymbol{m}} \left[\lambda^{\text{VR}} \cdot \ell^{\text{VR}} + \lambda^{\text{TL}} \cdot \ell^{\text{TL}} + \lambda^{\text{MASK}} \cdot \ell^{\text{MASK}} \right], \qquad (14)$$

where the expectation is taken with respect to random masking. Since the VR and TL subtasks share the same model and output representations, the final objective needs to balance different goals and is multi-tasking in nature. We provide a detailed ablation study in §4 to analyze the choice of weights.

3.5. Two-stage Inference with HAMMER

For the model inference of HAMMER, we perform two sequential stages, *i.e.*, video retrieval and temporal localization, to accomplish the task of moment localization in video corpus. For video retrieval, we use HAMMER and the linear regressor to compute pairwise compatibility scores as in Eq. (9) with respect to the text query $\boldsymbol{h}$ and all videos $\mathcal{V}$ in the corpus. Next, we perform temporal localization on the top ranked videos. Specifically, we predict the start and end frame with HAMMER to localize the temporal segment $\boldsymbol{s}$ following Eq. (5). Then we greedily label the frame with the maximum $p(t^{\text{s}}|\boldsymbol{v}, \boldsymbol{h})$ as the start frame and maximum $p(t^{\text{e}}|\boldsymbol{v}, \boldsymbol{h})$ as the end frame. Here we have an additional constraint to consider — the predicted end frame must appear after the start frame prediction. This two-stage inference reduces the complexity to $O(|\mathcal{V}| + \mathsf{N})$, which is significantly better comparing to the $O(|\mathcal{V}| \cdot \mathsf{N}^2)$ complexity of [8].

4. Experiments

In this section, we perform experiments with the proposed HAMMER model. We first introduce the datasets and setups of our experiments in §4.1. Next, we present the main results of the HAMMER model in §4.2, contrasting against a strong baseline FLAT as well as other existing methods. We then confuct a thorough ablation study in §4.3 to evaluate the importance of various design choices for the HAMMER model. Finally, we carry out qualitative analysis of our model to better demonstrate its behaviour.

4.1. Experimental Setups

Datasets We experiment on two popular MLVC datasets:

- **ActivityNet Captions** [19] contains ∼20K videos, each has 3.65 temporally localized query sentences on average. The mean video duration is 180 seconds and the average query length is 13.48 words, which spans over 36 seconds of the video. There are 10,009 videos for training and 4,917 videos for validation (val_1 split). We follow prior work [8, 15] to train our models and evaluate them on the val_1 split.
- **TVR** [20] contains ∼22K videos in total, of which ∼17.5K videos are in the training set and 2,180 are in the validation set. The dataset contains videos from movies, cartoons, and TV-shows. The videos are on average 76.2 seconds long and contain 5 temporally localized sentences per video. The moments in the videos are 9.1 seconds long and described by sentences containing 13.4 words on average. We make use of the subtitle (ASR) features together with the video feature in TVR dataset, following prior works [20, 21].

We make use of multiple popular choices of video features on these two datasets as existing literature [8, 15, 20], which includes the appearance-only features (ResNet152 [13] pre-trained on ImageNet [5]), spatio-temporal features (I3D [1] pre-trained on Kinetics [17]), and their combinations. We present the details of feature preparation in Suppl. Material.

Evaluation Metrics We use different evaluation metrics for different video understanding tasks:

- **Video Retrieval** (VR) We report Recall@k and Median Rank (MedR or MedRank) as the evaluation metrics for Video Retrieval as suggested in the literature.
- **Temporal Localization** (TL) We report both mean IoU (mIoU) and average precision with IoU=$\{0.3, 0.5, 0.7\}$ as the evaluation metrics. Here, IoU measures the Intersection over Union between the ground truth and predicted video segments, *i.e.*, the localization accuracy.
- **Moment Localization in Video Corpus** (MLVC) We use Recall@k with IoU=p for the main evaluation metrics [8, 20]. Specifically, we measure whether the correct localized segment exists in the top k of the ranked videos. Here, a localized segment is correct if it overlaps with the ground truth segment over an IoU of $\{0.5, 0.7\}$.

Baseline and the HAMMER Models In HAMMER, we use two encoders, *i.e.*, the frame and clip encoders, with multiple Transformer [31] layers to represent the visual (and

Table 1. MLVC Results on ActivityNet and TVR datasets

			IoU=0.5			IoU=0.7		
	Model & Feature		R1	R10	R100	R1	R10	R100
ActivityNet	MCN [14]	R	0.02	0.18	1.26	0.01	0.09	0.70
	CAL [8]	R	0.21	1.32	6.82	0.12	0.89	4.79
	FLAT	R	0.34	2.28	10.09	0.21	1.28	5.69
	HAMMER	R	0.51	3.29	12.01	0.30	1.87	6.94
	FLAT	I	2.57	13.07	30.66	1.51	7.69	17.67
	HAMMER	I	**2.94**	**14.49**	**32.49**	**1.74**	**8.75**	**19.08**
TVR	XML [20]	I+R	–	–	–	2.62	6.39	**22.00**
	HERO[1] [21]	I+R	–	–	–	2.98	10.65	18.25
	FLAT	I+R	8.45	21.14	30.75	4.61	11.29	16.24
	HAMMER	I+R	**9.19**	**21.28**	**31.25**	**5.13**	**11.38**	16.71

Table 2. VR results on ActivityNet Captions.

Model	R1	R10	R100	MedR↓
FLAT	5.37	29.14	71.64	29
HAMMER	**5.89**	**30.98**	**73.38**	**26**

Table 3. TL results on ActivityNet Captions.

Model	IoU=0.3	IoU=0.5	IoU=0.7	mIoU↑
FLAT	57.58	39.60	22.59	40.98
HAMMER	**59.18**	**41.45**	**24.27**	**42.68**

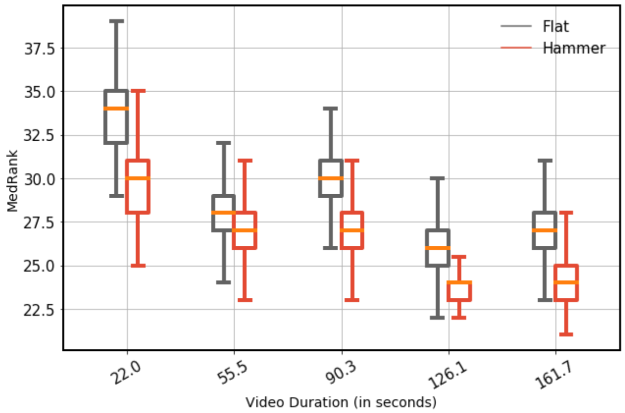

Figure 2. Comparison of Video Retrieval performances under different video duration. Results are reported in Median Rank (MedRank) on the ActivityNet Captions (**Lower is better**).

ASR) features as well as the text query features (details in Figure 1). Each encoder contains 1 layer of Transformer for visual input, 5 layers of Transformers for the text query input, and 1 layer of cross-modal Transformer between the visual and text query inputs. When ASR is provided (*i.e*., in TVR), we add one additional Transformer layer to incorporate the ASR input, with another cross-modal Transformer layer that cross-attends between the query input and ASR features. The processed ASR and visual features are concatenated. Meanwhile, we design a FLAT model as a strong baseline. The FLAT model has a similar architecture as HAMMER, except that it only uses the frame encoder to capture the visual (and ASR) features. We provide complete details about the architectural configurations and model optimization in the Suppl. Material.

4.2. MLVC Experiments

Main Results Table 1 presents a comparison between the proposed HAMMER and other methods on the two MLVC benchmarks. We observe that, irrespective of the feature types, HAMMER outperforms FLAT noticeably, which in turn outperforms most published results on both datasets. On ActivityNet, we observe that models using I3D features (denoted as I) outperform their counterparts with ResNet152 (denoted as R) features, by a significant margin. It indicates the importance of spatio-temporal feature representation in the MLVC tasks.

Meanwhile, we note that our FLAT model outperforms the baselines on the TVR dataset, which is mainly due to the introduction of the cross-modal Transformer between query and visual+ASR features (see §4.3 for a detailed study). On both datasets, HAMMER establishes the new state-of-the-art results for the MLVC task (without using additional data). This result shows a clear benefit of hierarchical structure modeling in video for the MLVC task.

Table 2 and 3 contrast HAMMER to the FLAT model in more details by comparing their performance on the tasks of video retrieval and temporal localization separately. The results are reported on the ActivityNet with models using the I3D features. In both cases, HAMMER achieves significantly better performance than the baseline FLAT model.

Comparing Models on Videos of Different Duration We discuss the potential reasons for HAMMER to outperform the FLAT model. Since HAMMER learns video representation at multiple granularities, we hypothesize that it should be able to focus on the task-relevant parts of a video without getting distracted by irrelevant parts. Specifically for the task of sentence-based video retrieval which requires matching the relevant frames in the video with the text query, HAMMER would be less sensitive to the presence of non-matching frames and hence be robust to the length of the video. To verify this, we analyze HAMMER's performance on videos with different lengths for the task of video retrieval and temporal localization.

We compare the performance of HAMMER and FLAT on videos with different durations for the video retrieval task in Fig. 2. The metric used for comparison is the median rank where lower numbers indicate better performance. Firstly, it can be observed that while the performance of FLAT model

[1] We compare against their model without large-scale pre-training for fair comparison.

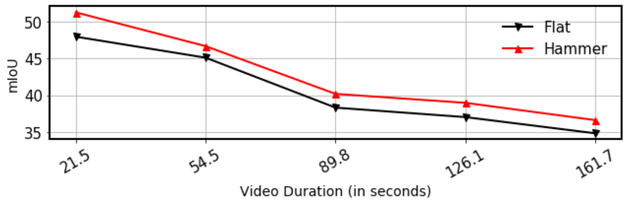

Figure 3. Comparison of Temporal Localization performances under different video duration. Results are reported in Mean IoU (mIoU) on the ActivityNet Captions (**Higher is better**).

is inconsistent (e.g., performance on longest videos is worse than second-to-longest videos), the HAMMER model's performance consistently improves with the length of the video. Secondly, the performance of the HAMMER model is best for the longest videos in the dataset. Finally, while both the models perform sub-optimally on the shortest videos, HAMMER still outperforms FLAT for those videos.

We further compare the temporal localization performance of HAMMER and FLAT models in Fig. 3. The results are reported using mean IoU, where higher numbers indicate better performance. It shows that HAMMER constantly achieves higher performance than FLAT across all videos irrespective of their length.

Overall, the analysis shows clear advantage of using HAMMER over FLAT which is especially profound for longer videos, hence supporting our central modeling argument.

4.3. Ablation Studies and Analyses

In this section, we evaluate the effectiveness of learning objectives and various design choices for HAMMER. We note that all the ablation studies in this section are conducted on ActivityNet Captions using the I3D features.

4.3.1 Learning Objectives

As aforementioned, HAMMER is optimized with three objectives jointly namely video retrieval (VR), temporal localization (TL), and masked language modeling (MLM). We study the contribution of different objectives, reported in Table 4. It is worth to note that here we differentiate the MLM objective applied to the frame encoder (denote as FM) and the clip encoder (denote as CM). Firstly, the objectives of VR and TL are complementary to each other and jointly optimizing the two surpasses the single task performance on both the tasks simultaneously. Secondly, CM and FM applied individually benefits both VR and TL tasks with their usage in unison resulting in best performance. This verifies the effectiveness of MLM objective to improve the text representation. Finally, the best performance is achieved by combining all the objectives, hence proving the complimentary nature of all of them.

Weights of Different Objectives We also conduct detailed experiments to investigate the influence of different objectives' weights (λ^{VR}, λ^{TL}, and λ^{MLM}). Table 5 shows that it is important to balance the weights between VR and TL. The best performance is achieved when the weight for VR and TL is set to $1:5$. For MLM, we find that the best loss weight is 0.1, and thus use this value through all our experiments.

Table 4. Ablation study on sub-tasks (VR=Video Retrieval, TL=Temporal Localization, FM=Frame MLM, CM=Clip MLM)

Task				Video Retrieval			Temporal Localization		
VR	TL	FM	CM	R1	R10	R100	IoU=0.5	IoU=0.7	mIoU
✓				4.93	29.02	72.15	–	–	–
✓		✓		5.52	30.53	73.02	–	–	–
✓			✓	5.45	30.45	73.24	–	–	–
✓		✓	✓	5.67	30.20	72.67	–	–	–
	✓			–	–	–	39.02	22.74	40.28
	✓	✓		–	–	–	39.27	22.04	40.30
	✓		✓	–	–	–	39.13	22.38	40.51
	✓	✓	✓	–	–	–	39.16	22.82	40.64
✓	✓			5.22	30.22	72.70	40.59	23.70	42.01
✓	✓	✓		5.57	30.97	73.09	41.17	24.04	42.45
✓	✓		✓	5.85	30.82	**73.54**	41.30	23.94	42.43
✓	✓	✓	✓	**5.89**	**30.98**	73.38	**41.45**	**24.27**	**42.68**

Table 5. Ablation study on task weights (VR=Video Retrieval, TL=Temporal Localization, MLM=Masked Language Model)

			IoU=0.5			IoU=0.7		
λ^{VR}	λ^{TL}	λ^{MLM}	R1	R10	R100	R1	R10	R100
1.0	0.1	0.1	1.65	9.18	20.81	0.87	4.88	10.50
1.0	0.5	0.1	2.15	10.75	23.41	1.10	5.68	12.16
1.0	1.0	0.1	2.02	10.95	24.74	1.10	6.07	13.12
1.0	5.0	0.1	**2.94**	**14.49**	**32.49**	**1.74**	**8.75**	**19.08**
1.0	10.0	0.1	2.35	14.25	31.84	1.42	8.53	18.76

Table 6. Ablation study on Cross-modal Transformer

Model	X-Modal	IoU=0.5			IoU=0.7		
		R1	R10	R100	R1	R10	R100
HAMMER	✗	1.38	8.89	26.35	0.84	5.08	15.27
	✓	**2.94**	**14.49**	**32.49**	**1.74**	**8.75**	**19.08**

4.3.2 Evaluate Design Choices of the HAMMER

We study the importance of a few design choices in the HAMMER model. Specifically, we evaluate the following:

- Effect of the *cross-modal Transformer* layer
- Effect of different *clip lengths* for the clip representation
- Effect of *parameter sharing* for frame and clip encoders
- Effect of an additional clip-level *position embedding*

We present the results and discussion on these experiments in the following paragraphs.

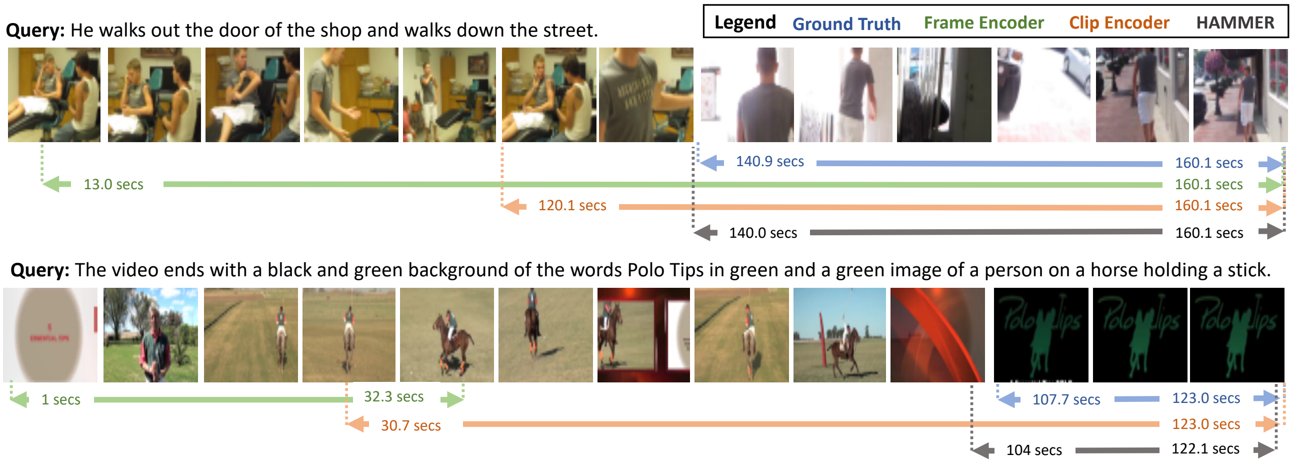
Query: He walks out the door of the shop and walks down the street.
Legend Ground Truth Frame Encoder Clip Encoder HAMMER
140.9 secs
160.1 secs
13.0 secs
160.1 secs
120.1 secs
160.1 secs
140.0 secs
160.1 secs
Query: The video ends with a black and green background of the words Polo Tips in green and a green image of a person on a horse holding a stick.
1 secs
32.3 secs
30.7 secs
107.7 secs
123.0 secs
123.0 secs
104 secs
122.1 secs

Figure 4. Illustration of temporal localization using different hierarchies of HAMMER as well as the final HAMMER model

Table 7. Ablation study on different clip lengths

Model	Clip Length	IoU=0.5			IoU=0.7		
		R1	R10	R100	R1	R10	R100
HAMMER	16	2.70	14.06	31.85	1.63	8.16	18.60
	32	**2.94**	14.49	**32.49**	**1.74**	8.75	**19.08**
	64	2.78	**14.69**	32.08	1.70	**9.00**	18.71

Table 8. Ablation study on weight sharing

Model	Weight Sharing	IoU=0.5			IoU=0.7		
		R1	R10	R100	R1	R10	R100
HAMMER	✗	**2.94**	**14.49**	**32.49**	**1.74**	**8.75**	**19.08**
	✓	2.89	14.17	30.31	1.69	8.05	17.24

Table 9. Ablation study on clip position embeddings

Model	Clip Position	IoU=0.5			IoU=0.7		
		R1	R10	R100	R1	R10	R100
HAMMER	✗	2.82	14.39	32.01	1.76	8.59	18.63
	✓	**2.94**	**14.49**	**32.49**	**1.74**	**8.75**	**19.08**

Cross-Modal Transformer is Essential. Both frame and clip encoders contain one layer of cross-modal (X-modal) Transformer between text query and video inputs. To verify its effectiveness, we compare with an ablation model without this layer. Table 6 shows almost 100% relative improvement in the R1, R10 metrics when using the X-modal Transformer, proving it is essential to the success of HAMMER.

Optimal Length of the Clip-Level Representation. In HAMMER, recall that a frame encoder takes a clip of fixed length of frames and outputs a clip-level representation. Here, we examine the performance under different lengths of clips, summarized in Table 7. Overall, we observe that the model's performance is robust to the clip length chosen for the experiments, and 32 is the optimal length for the clip representation (with max video length of 128).

Parameter Sharing between Frame/Clip Encoders. We also consider whether the frame encoder and the clip encoder in the HAMMER model may share the same set of parameters, as weight sharing could regularize the model capacity and therefore improve the generalization performance. Table 8 indicates that, however, untying the encoder weights achieves slightly better performance, potentially thanks to its greater flexibility.

Position Embedding for Clip Encoder. Position embedding is an important model input as it indicates the temporal boundary of each video frame segment. In the HAMMER model, since we also have a clip encoder that takes the aggregated "Clip CLS" token as input, it is natural to ask if we need a position encoding for each clip representation. Thus, we compare two models, with and without additional clip position encoding. Table 9 shows that clip position embedding is indeed important to achieve superior performance.

4.3.3 Qualitative Visualization

To better understand the behavior of the HAMMER model, we demonstrate a couple of examples of temporal localization. Figure 4 lists predicted spans from the frame and clip encoder as well as from the entire HAMMER. In both examples, we observe that the frame encoder of HAMMER makes an incorrect prediction of the temporal timestamps, but then corrected by the prediction from the clip encoder. Overall, HAMMER makes more accurate predictions with respect to the ground-truth video segment.

5. Conclusion

In this paper, we propose a hierarchical multi-modal encoder (HAMMER) that captures video dynamics in three scales of granularity, frame, clip, and video. By hierarchically modeling videos, HAMMER achieves significantly better performance than the baseline approaches on moment localization task in video corpus, and further establishes new state-of-the-art on two challenging datasets, ActivityNet captions and TVR. Extensive studies verify the effectiveness of the proposed architectures and learning objectives.

References

[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR*, 2017. 5

[2] Jingyuan Chen, Xinpeng Chen, Lin Ma, Zequn Jie, and Tat-Seng Chua. Temporally grounding natural sentence in video. In *EMNLP*, 2018. 2

[3] Jiacheng Chen, Hexiang Hu, Hao Wu, Yuning Jiang, and Changhu Wang. Learning the best pooling strategy for visual semantic embedding. *arXiv:2011.04305*, 2020. 2

[4] Long Chen, Chujie Lu, Siliang Tang, Jun Xiao, Dong Zhang, Chilie Tan, and Xiaolin Li. Rethinking the bottom-up framework for query-based video localization. In *AAAI*, 2020. 2

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*, 2018. 4

[7] Jianfeng Dong, Xirong Li, and Cees G. M. Snoek. Word2VisualVec: Cross-media retrieval by visual feature prediction. *arXiv:1604.06838*, 2016. 2

[8] Victor Escorcia, Mattia Soldan, Josef Sivic, Bernard Ghanem, and Bryan Russell. Temporal localization of moments in video collections with natural language. *arXiv:1907.12763*, 2019. 1, 2, 5, 6

[9] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612*, 2017. 2

[10] Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. Entities as experts: Sparse memory access with entity supervision. *arXiv:2004.07202*, 2020. 3

[11] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013. 2

[12] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. TALL: Temporal activity localization via language query. In *ICCV*, 2017. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[14] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *ICCV*, 2017. 2, 6

[15] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with temporal language. In *EMNLP*, 2018. 5

[16] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020. 3

[17] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv:1705.06950*, 2017. 5

[18] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv:1411.2539*, 2014. 2

[19] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, 2017. 2, 5

[20] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. TVR: A large-scale dataset for video-subtitle moment retrieval. *arXiv:2001.09099*, 2020. 1, 2, 3, 5, 6

[21] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+language omni-representation pre-training, 2020. 5, 6

[22] Meng Liu, Xiang Wang, Liqiang Nie, Qi Tian, Baoquan Chen, and Tat-Seng Chua. Cross-modal moment localization in videos. In *ACM MM*, 2018. 2

[23] Chujie Lu, Long Chen, Chilie Tan, Xiaolin Li, and Jun Xiao. DEBUG: A dense bottom-up grounding approach for natural language video localization. In *EMNLP*, 2019. 2

[24] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. VilBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019. 4

[25] 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. 2

[26] Cisco Visual Networking. Cisco global cloud index: Forecast and methodology, 2016–2021. *White paper. Cisco Public, San Jose*, 2016. 1

[27] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, 2016. 2

[28] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics*, 1:25–36, 2013. 2

[29] Dian Shao, Yu Xiong, Yue Zhao, Qingqiu Huang, Yu Qiao, and Dahua Lin. Find and focus: Retrieve and localize video events with natural language queries. In *ECCV*, 2018. 2

[30] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. VideoBERT: A joint model for video and language representation learning. In *ICCV*, 2019. 4

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 5

[32] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko.

Sequence to sequence – video to text. In *ICCV*, 2015. 2

[33] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. *arXiv:1412.4729*, 2014. 2

[34] Huijuan Xu, Kun He, Bryan Plummer, Leonid Sigal, Stan Sclaroff, and Kate Saenko. Multilevel language and vision integration for text-to-clip retrieval. In *AAAI*, 2019. 2

[35] Huijuan Xu, Kun He, Leonid Sigal, Stan Sclaroff, and Kate Saenko. Text-to-clip video retrieval with early fusion and re-captioning. *arXiv:1804.05113*, 2018. 2

[36] Ran Xu, Caiming Xiong, Wei Chen, and Jason J. Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *AAAI*, 2015. 2

[37] Yitian Yuan, Lin Ma, Jingwen Wang, Wei Liu, and Wenwu Zhu. Semantic conditioned dynamic modulation for temporal sentence grounding in videos. In *NeurIPS*, 2019. 2

[38] 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. 2

[39] Bowen Zhang, Hexiang Hu, and Fei Sha. Cross-modal and hierarchical modeling of video and text. In *ECCV*, 2018. 2, 4