Localizing Moments in Video with Natural Language

Lisa Anne Hendricks[1][*], Oliver Wang[2], Eli Shechtman[2], Josef Sivic[2,3][*], Trevor Darrell[1], Bryan Russell[2]
[1]UC Berkeley, [2]Adobe Research, [3]INRIA
https://people.eecs.berkeley.edu/~lisa_anne/didemo.html

Text query: The little girl jumps back up after falling.

Figure 1: We consider localizing moments in video with natural language and demonstrate that incorporating local and global video features is important for this task. To train and evaluate our model, we collect the Distinct Describable Moments (DiDeMo) dataset which consists of over 40,000 pairs of localized video moments and corresponding natural language.

Abstract

We consider retrieving a specific temporal segment, or moment, from a video given a natural language text description. Methods designed to retrieve whole video clips with natural language determine what occurs in a video but not when. To address this issue, we propose the Moment Context Network (MCN) which effectively localizes natural language queries in videos by integrating local and global video features over time. A key obstacle to training our MCN model is that current video datasets do not include pairs of localized video segments and referring expressions, or text descriptions which uniquely identify a corresponding moment. Therefore, we collect the Distinct Describable Moments (DiDeMo) dataset which consists of over 10,000 unedited, personal videos in diverse visual settings with pairs of localized video segments and referring expressions. We demonstrate that MCN outperforms several baseline methods and believe that our initial results together with the release of DiDeMo will inspire further research on localizing video moments with natural language.

1. Introduction

Consider the video depicted in Figure 1, in which a little girl jumps around, falls down, and then gets back up to start jumping again. Suppose we want to refer to a particular temporal segment, or moment, from the video, such as

when the girl resiliently begins jumping again after she has fallen. Simply referring to the moment via an action, object, or attribute keyword may not uniquely identify it. For example, important objects in the scene, such as the girl, are present in each frame. Likewise, recognizing all the frames in which the girl is jumping will not localize the moment of interest as the girl jumps both before and after she has fallen. Rather than being defined by a single object or activity, the moment may be defined by when and how specific actions take place *in relation* to other actions. An intuitive way to refer to the moment is via a natural language phrase, such as "the little girl jumps back up after falling".

Motivated by this example, we consider localizing moments in video with natural language. Specifically, given a video and text description, we identify start and end points in the video which correspond to the given text description. This is a challenging task requiring both language and video understanding, with important applications in video retrieval, such as finding particular moments from a long personal holiday video, or desired B-roll stock video footage from a large video library (e.g., Adobe Stock[1], Getty[2], Shutterstock[3]).

Existing methods for natural language based video retrieval [24, 51, 46] retrieve an entire video given a text string but do not identify *when* a moment occurs within a video. To localize moments within a video we propose to learn a joint video-language model in which referring expressions and video features from corresponding moments are close

---

[1]https://stock.adobe.com
[2]http://www.gettyimages.com
[3]https://www.shutterstock.com

in a shared embedding space. However, in contrast to whole video retrieval, we argue that in addition to video features from a specific moment, global video context and knowing when a moment occurs within a longer video are important cues for moment retrieval. For example, consider the text query "The man on the stage comes closest to the audience". The term "closest" is relative and requires temporal context to properly comprehend. Additionally, the temporal position of a moment in a longer video can help localize the moment. For the text query "The biker starts the race", we expect moments earlier in the video in which the biker is racing to be closer to the text query than moments at the end of the video. We thus propose the Moment Context Network (MCN) which includes a global video feature to provide temporal context and a temporal endpoint feature to indicate when a moment occurs in a video.

A major obstacle when training our model is that current video-language datasets do not include natural language which can uniquely localize a moment. Additionally, datasets like [20, 28] are small and restricted to specific domains, such as dash-cam or cooking videos, while datasets [5, 32, 45, 50] sourced from movies and YouTube are frequently edited and tend to only include entertaining moments (see [38] for discussion). We believe the task of localizing moments with natural language is particularly interesting in unedited videos which tend to include uneventful video segments that would generally be cut from edited videos. Consequently, we desire a dataset which consists of distinct moments from unedited video footage paired with descriptions which can uniquely localize each moment, analogous to datasets that pair distinct image regions with descriptions [17, 22].

To address this problem, we collect the Distinct Describable Moments (DiDeMo) dataset which includes distinct video moments paired with descriptions which uniquely localize the moment in the video. Our dataset consists of over 10,000 unedited videos with 3-5 pairs of descriptions and distinct moments per video. DiDeMo is collected in an open-world setting and includes diverse content such as pets, concerts, and sports games. To ensure that descriptions are referring and thus uniquely localize a moment, we include a validation step inspired by [17].

Contributions. We consider the problem of localizing moments in video with natural language in a challenging open-world setting. We propose the Moment Context Network (MCN) which relies on local and global video features. To train and evaluate our model, we collect the Distinct Describable Moments (DiDeMo) dataset which consists of over 40,000 pairs of referring descriptions and localized moments in unedited videos.

2. Related Work

Localizing moments in video with natural language is related to other vision tasks including video retrieval, video summarization, video description and question answering, and natural language object retrieval. Though large scale datasets have been collected for each of these tasks, none fit the specific requirements needed to learn how to localize moments in video with natural language.

Video Retrieval with Natural Language. Natural language video retrieval methods aim to retrieve a specific video given a natural language query. Current methods [24, 46, 51] incorporate deep video-language embeddings similar to image-language embeddings proposed by [8, 40]. Our method also relies on a joint video-language embedding. However, to identify when events occur in a video, our video representation integrates local and global video features as well as temporal endpoint features which indicate when a candidate moment occurs within a video.

Some work has studied retrieving temporal segments within a video in constrained settings. For example, [43] considers retrieving video clips from a home surveillance camera using text queries which include a fixed set of spatial prepositions ("across" and "through") whereas [20] considers retrieving temporal segments in 21 videos from a dashboard car camera. In a similar vein, [1, 4, 36] consider aligning textual instructions to videos. However, methods aligning instructions to videos are restricted to structured videos as they constrain alignment by instruction ordering. In contrast, we consider localizing moments in an unconstrained open-world dataset with a wide array of visual concepts. To effectively train a moment localization model, we collect DiDeMo which is unique because it consists of paired video moments and referring expressions.

Video Summarization. Video summarization algorithms isolate temporal segments in a video which include important/interesting content. Though most summarization algorithms do not include textual input ([3, 9, 10, 52, 53]), some use text in the form of video titles [21, 41] or user queries in the form of category labels to guide content selection [37]. [54] collects textual descriptions for temporal video chunks as a means to evaluate summarization algorithms. However, these datasets do not include referring expressions and are limited in scope which makes them unsuitable for learning moment retrieval in an open-world setting.

Video Description and Question Answering (QA). Video description models learn to generate textual descriptions of videos given video-description pairs. Contemporary models integrate deep video representations with recurrent language models [25, 31, 47, 48, 56]. Additionally, [42] proposed a video QA dataset which includes question/answer pairs aligned to video shots, plot synopsis, and subtitles.

YouTube and movies are popular sources for joint video-

language datasets. Video description datasets collected from YouTube include descriptions for short clips of longer YouTube videos [5, 50]. Other video description datasets include descriptions of short clips sourced from full length movies [32, 45]. However, though YouTube clips and movie shots are sourced from longer videos, they are not appropriate for localizing distinct moments in video for two reasons. First, descriptions about selected shots and clips are not guaranteed to be referring. For example, a short YouTube video clip might include a person talking and the description like "A woman is talking". However, the entire video could consist of a woman talking and thus the description does not uniquely refer to the clip. Second, many YouTube videos and movies are edited, which means "boring" content which may be important to understand for applications like retrieving video segments from personal videos might not be present.

Natural Language Object Retrieval. Natural language object retrieval [14, 22] can be seen as an analogous task to ours, where natural language phrases are localized spatially in images, rather than temporally in videos. Despite similarities to natural language object retrieval, localizing video moments presents unique challenges. For example, it often requires comprehension of temporal indicators such as "first" as well as a better understanding of activities. Datasets for natural language object retrieval include *referring* expressions which can uniquely localize a specific location in a image. Descriptions in DiDeMo uniquely localize distinct moments and are thus also referring expressions.

Language Grounding in Images and Videos. [27, 29, 40] tackle the task of object grounding in which sentence fragments in a description are localized to specific image regions. Work on language grounding in video is much more limited. Language grounding in video has focused on spatially grounding objects and actions in a video [20, 55], or aligning textual phrases to temporal video segments [28, 43]. However prior methods in both these areas ([43, 55]) severely constrain natural language vocabulary (e.g., [55] only considers four objects and four verbs) and consider constrained visual domains in small datasets (e.g., 127 videos from a fixed laboratory kitchen [28] and [20] only includes 520 sentences). In contrast, DiDeMo offers a unique opportunity to study temporal language grounding in an open-world setting with a diverse set of objects, activities, and attributes.

3. Moment Context Network

Our moment retrieval model effectively localizes natural language queries in longer videos. Given input video frames $v = \{v_t\}$, where $t \in \{0, \ldots, T-1\}$ indexes time, and a proposed temporal interval, $\hat{\tau} = \tau_{start} : \tau_{end}$, we extract visual temporal context features which encode the

Figure 2: Our Moment Context Network (MCN) learns a shared embedding for video temporal context features and LSTM language features. Our video temporal context features integrate local video features, which reflect what occurs during a specific moment, global features, which provide context for the specific moment, and temporal endpoint features which indicate when a moment occurs in a video. We consider both appearance and optical flow input modalities, but for simplicity only show the appearance input modality here.

video moment by integrating both local features and global video context. Given a sentence s we extract language features using an LSTM [12] network. At test time our model optimizes the following objective

$$\hat{\tau} = \underset{\tau}{\arg\min}\, D_\theta(s, v, \tau), \qquad (1)$$

where D_θ is a joint model over the sentence s, video v, and temporal interval τ given model parameters θ (Figure 2).

Visual Temporal Context Features. We encode video moments into visual temporal context features by integrating local video features, which reflect what occurs within a specific moment, global video features, which provide context for a video moment, and temporal endpoint features, which indicate when a moment occurs within a longer video. To construct local and global video features, we first extract high level video features using a deep convolutional network for each video frame, then average pool video features across a specific time span (similar to features employed by [48] for video description and [46] for whole video retrieval). Local features are constructed by pooling features within a specific moment and global features are constructed by averaging over all frames in a video.

When a moment occurs in a video can indicate whether or not a moment matches a specific query. To illustrate, consider the query "the bikers start the race." We expect moments closer to the beginning of a video in which bikers are racing to be more similar to the description than moments at the end of the video in which bikers are racing.

To encode this temporal information, we include temporal endpoint features which indicate the start and endpoint of a candidate moment (normalized to the interval $[0, 1]$). We note that our global video features and temporal endpoint features are analogous to global image features and spatial context features frequently used in natural language object retrieval [14, 22].

Localizing video moments often requires localizing specific activities (like "jump" or "run"). Therefore, we explore two sources of visual input modalities; appearance or RGB frames (v_t) and optical flow frames (f_t). We extract fc_7 features from RGB frames using VGG [39] pre-trained on ImageNet [35]. We expect these features to accurately identify specific objects and attributes in video frames. Likewise, we extract optical flow features from the penultimate layer from a competitive activity recognition model [49]. We expect these features to help localize moments which require understanding action.

Temporal context features are extracted by inputting local video features, global video features, and temporal endpoint features into a two layer neural network with ReLU nonlinearities (Figure 2 top). Separate weights are learned when extracting temporal context features for RGB frames (denoted as P_θ^V) and optical flow frames (denoted as P_θ^F).

Language Features. To capture language structure, we extract language features using a recurrent network (specifically an LSTM [12]). After encoding a sentence with an LSTM, we pass the last hidden state of the LSTM through a single fully-connected layer to yield embedded feature P_θ^L. Though our dataset contains over 40,000 sentences, it is still small in comparison to datasets used for natural language object retrieval (e.g., [17, 22]). Therefore, we find that representing words with dense word embeddings (specifically Glove [26]) as opposed to one-hot encodings yields superior results when training our LSTM.

Joint Video and Language Model. Our joint model is the sum of squared distances between embedded appearance, flow, and language features

$$D_\theta(s, v, \tau) = |P_\theta^V(v, \tau) - P_\theta^L(s)|^2 + \eta |P_\theta^F(f, \tau) - P_\theta^L(s)|^2, \quad (2)$$

where η is a tunable (via cross validation) "late fusion" scalar parameter. η was set to 2.33 via ablation studies.

Ranking Loss for Moment Retrieval. We train our model with a ranking loss which encourages referring expressions to be closer to corresponding moments than negative moments in a shared embedding space. Negative moments used during training can either come from different segments within the same video (intra-video negative moments) or from different videos (inter-video negative moments). Revisiting the video depicted in Figure 1, given a phrase "the little girl jumps back up after falling" many intra-video negative moments include concepts mentioned

in the phrase such as "little girl" or "jumps". Consequently, our model must learn to distinguish between subtle differences within a video. By comparing the positive moment to the intra-video negative moments, our model can learn that localizing the moment corresponding to "the little girl jumps back up after falling" requires more than just recognizing an object (the girl) or an action (jumps). For training example i with endpoints τ_i, we define the following intra-video ranking loss

$$\mathcal{L}_i^{intra}(\theta) = \sum_{n \in \Gamma \setminus \tau^i} \mathcal{L}^R \left(D_\theta(s^i, v^i, \tau^i), D_\theta(s^i, v^i, n) \right), \tag{3}$$

where $\mathcal{L}^R(x, y) = \max(0, x - y + b)$ is the ranking loss, Γ are all possible temporal video intervals, and b is a margin. Intuitively, this loss encourages text queries to be closer to a corresponding video moment than all other possible moments from the same video.

Only comparing moments within a single video means the model must learn to differentiate between subtle differences without learning how to differentiate between broader semantic concepts (e.g., "girl" vs. "sofa"). Hence, we also compare positive moments to inter-video negative moments which generally include substantially different semantic content. When selecting inter-video negative moments, we choose negative moments which have the same start and end points as positive moments. This encourages the model to differentiate between moments based on semantic content, as opposed to when the moment occurs in the video. During training we do not verify that inter-video negatives are indeed true negatives. However, the language in our dataset is diverse enough that, in practice, we observe that randomly sampled inter-video negatives are generally true negatives. For training example i, we define the following inter-video ranking loss

$$\mathcal{L}_i^{inter}(\theta) = \sum_{j \neq i} \mathcal{L}^R \left(D_\theta(s^i, v^i, \tau^i), D_\theta(s^i, v^j, \tau^i) \right). \tag{4}$$

This loss encourages text queries to be closer to corresponding video moments than moments outside the video, and should thus learn to differentiate between broad semantic concepts. Our final inter-intra video ranking loss is

$$\mathcal{L}(\theta) = \lambda \sum_i \mathcal{L}_i^{intra}(\theta) + (1 - \lambda) \sum_i \mathcal{L}_i^{inter}(\theta), \tag{5}$$

where λ is a weighting parameter chosen through cross-validation.

4. The DiDeMo Dataset

A major challenge when designing algorithms to localize moments with natural language is that there is a dearth of large-scale datasets which consist of referring expressions and localized video moemnts. To mitigate this issue,

The woman squats down and pets the dog

The woman stands and begins walking away from her dog

The dog begins to follow the woman as she leads it through the training course

A black and white dog runs through an indoor agility course

A dog jumps over two white jumps

A cat walks over two boxes

An orange cat walks out of a box

There is a different cat in each of the two boxes

A black cat walks out of a box that the orange cat used to be in

Black cat walks into right hand side box

Figure 3: Example videos and annotations from our Distinct Describable Moments (DiDeMo) dataset. Annotators describe moments with varied language (e.g., "A cat walks over two boxes" and "An orange cat walks out of a box"). Videos with multiple events (top) have annotations which span all five-second segments. Other videos have segments in which no distinct event takes place (e.g., the end of the bottom video in which no cats are moving).

we introduce the Distinct Describable Moments (DiDeMo) dataset which includes over 10,000 25-30 second long personal videos with over 40,000 localized text descriptions. Example annotations are shown in Figure 3.

4.1. Dataset Collection

To ensure that each description is paired with a single distinct moment, we collect our dataset in two phases (similar to how [17] collected text to localize image regions). First, we asked annotators to watch a video, select a moment, and describe the moment such that another user would select the same moment based on the description. Then, descriptions collected in the first phase are validated by asking annotators to watch videos and mark moments that correspond to collected descriptions.

Harvesting Personal Videos. We randomly select over 14,000 videos from YFCC100M [44] which contains over 100,000 Flickr videos with a Creative Commons License. To ensure harvested videos are unedited, we run each video through a shot detector based on the difference of color histograms in adjacent frames [23] then manually filter videos which are not caught. Videos in DiDeMo represent a diverse set of real-world videos, which include interesting, distinct moments, as well as uneventful segments which might be excluded from edited videos.

Video Interface. Localizing text annotations in video is difficult because the task can be ambiguous and users must digest a 25-30s video before scrubbing through the video to mark start and end points. To illustrate the inherent ambiguity of our task, consider the phrase "The woman leaves the room." Some annotators may believe this moment begins as soon as the woman turns towards the exit, whereas others may believe the moment starts as the woman's foot first crosses the door threshold. Both annotations are valid, but

result in large discrepancies between start and end points.

To make our task less ambiguous and speed up annotation, we develop a user interface in which videos are presented as a timeline of temporal segments. Each segment is displayed as a gif, which plays at 2x speed when the mouse is hovered over it. Following [54], who collected localized text annotations for summarization datasets, we segment our videos into 5-second segments. Users select a moment by clicking on all segments which contain the moment. To validate our interface, we ask five users to localize moments in ten videos using our tool and a traditional video scrubbing tool. Annotations with our gif-based tool are faster to collect (25.66s vs. 38.48s). Additionally, start and end points marked using the two different tools are similar. The standard deviation for start and end points marked when using the video scrubbing tool (2.49s) is larger than the average difference in start and end points marked using the two different tools (2.45s).

Moment Validation. After annotators describe a moment, we ask three additional annotators to localize the moment given the text annotation and the same video. To accept a moment description, we require that at least three out of four annotators (one describer and three validators) be in agreement. We consider two annotators to agree if one of the start *or* end point differs by at most one gif.

4.2. DiDeMo Summary

Table 1 compares our Distinct Describable Moments (DiDeMo) dataset to other video-language datasets. Though some datasets include temporal localization of natural language, these datasets do not include a verification step to ensure that descriptions refer to a single moment. In contrast, our verification step ensuring that descriptions in DiDeMo are *referring expressions*, meaning that they refer

Dataset	# Videos/# Clips	# Sentences	Video Source	Domain	Temporal Localization	Un-Edited	Referring Expressions
YouCook [7]	88/-	2,668	YouTube	Cooking			
Charades [38]	10,000/-	16,129	Homes	Daily activities		✓	
TGIF [19]	100,000 /-	125,781	Tumblr GIFs	Open			
MSVD [5]	1,970/1,970	70,028	YouTube	Open	✓		
MSR-VTT [50]	7,180/10,000	200,000	YouTube	Open	✓		
LSMDC 16 [33]	200/128,085	128,085	Movie	Open	✓		
TV Dataset [54]	4/1,034	1,034	TV Shows	TV Shows	✓		
KITTI [20]	21/520	520	Car Camera	Driving	✓	✓	
TACoS [28, 34]	123/7,206	18,227	Lab Kitchen	Cooking	✓	✓	
TACoS multi-level[30]	185/14.105	52,593	Lab Kitchen	Cooking	✓	✓	
UT Egocentric [54]	4/11,216	11,216	Egocentric	Daily Activities	✓	✓	
Disneyland [54]	8/14,926	14,916	Egocentric	Disneyland	✓	✓	
DiDeMo	10,464/26,892	40,543	Flickr	Open	✓	✓	✓

Table 1: Comparison of DiDeMo to other video-language datasets. DiDeMo is unique because it includes a validation step ensuring that descriptions are referring expressions.

to a specific moment in a video.

Vocabulary. Because videos are curated from Flickr, DiDeMo reflects the type of content people are interested in recording and sharing. Consequently, DiDeMo is human-centric with words like "baby", "woman", and "man" appearing frequently. Since videos are randomly sampled, DiDeMo has a long tail with words like "parachute" and "violin", appearing infrequently (28 and 38 times).

Important, distinct moments in a video often coincide with specific camera movements. For example, "the camera pans to a group of friends" or "zooms in on the baby" can describe distinct moments. Many moments in personal videos are easiest to describe in reference to the viewer (e.g., "the little boy runs towards the camera"). In contrast to other dataset collection efforts [5], we allow annotations to reference the camera, and believe such annotations may be helpful for applications like text-assisted video editing.

Table 2 contrasts the kinds of words used in DiDeMo to two natural language object retrieval datasets [17, 22] and two video description datasets [33, 50]. The three left columns report the percentage of sentences which include camera words (e.g., "zoom", "pan", "cameraman"), temporal indicators (e.g., "after" and "first"), and spatial indicators (e.g., "left" and "bottom"). We also compare how many words belong to certain parts of speech (verb, noun, and adjective) using the natural language toolkit part-of-speech tagger [2]. DiDeMo contains more sentences with temporal indicators than natural language object retrieval and video description datasets, as well as a large number of spatial indicators. DiDeMo has a higher percentage of verbs than natural language object retrieval datasets, suggesting understanding action is important for moment localization in video.

	% Sentences			% Words		
	Camera	Temp.	Spatial	Verbs	Nouns	Adj.
ReferIt [17]	0.33	1.64	43.13	5.88	52.38	11.54
RefExp [22]	1.88	1.00	15.11	8.97	36.26	11.82
MSR-VTT [50]	2.10	2.03	1.24	18.77	36.95	5.12
LSMDC 16 [33]	1.09	7.58	1.49	13.71	37.44	3.99
DiDeMo	19.69	18.42	11.62	16.06	35.26	7.89

Table 2: DiDeMo contains more camera and temporal words than natural language object recognition datasets [17, 22] or video description datasets [50, 33]. Additionally, verbs are more common in DiDeMo than in natural language object retrieval datasets suggesting natural language moment retrieval relies more heavily on recognizing actions than natural language object retrieval.

Annotated Time Points. Annotated segments can be any contiguous set of gifs. Annotators generally describe short moments with 72.34% of descriptions corresponding to a single gif and 22.26% corresponding to two contiguous gifs. More annotated moments occur at the beginning of a video than the end. This is unsurprising as people generally choose to begin filming a video when something interesting is about to happen. In 86% of videos annotators described multiple distinct moments with an average of 2.57 distinct moments per video.

5. Evaluation

In this section we report qualitative and quantitative results on DiDeMo. First, we describe our evaluation criteria and then evaluate against baseline methods.

Metrics: Accounting for Human Variance. Our model ranks candidate moments in a video based on how well

Baseline Comparison (Test Set)			
Model	Rank@1	Rank@5	mIoU
1 Upper Bound	74.75	100.00	96.05
2 Chance	3.75	22.50	22.64
3 Moment Frequency Prior	19.40	66.38	26.65
4 CCA	18.11	52.11	37.82
5 Natural Lang. Obj. Retrieval [14]	16.20	43.94	27.18
6 Natural Lang. Obj. Retrieval [14] (re-trained)	15.57	48.32	30.55
7 MCN (ours)	**28.10**	**78.21**	**41.08**
Ablations (Validation Set)			
8 LSTM-RGB-local	13.10	44.82	25.13
9 LSTM-Flow-local	18.35	56.25	31.46
10 LSTM-Fusion-local	18.71	57.47	32.32
11 LSTM-Fusion + global	19.88	62.39	33.51
12 LSTM-Fusion + global + tef (MCN)	**27.57**	**79.69**	**41.70**

Table 3: Our Moment Context Network (MCN) outperforms baselines (rows 1-6) on our test set. We show ablation studies on our validation set in rows 8-12. Both flow and RGB modalities are important for good performance (rows 8-10). Global video features and temporal endpoint features (tef) both lead to better performance (rows 10-12).

they match a text description. Candidate moments come from the temporal segments defined by the gifs used to collect annotations. A 30 second video will be broken into six five-second gifs. Moments can include any contiguous set of gifs, so a 30-second video contains 21 possible moments. We measure the performance of each model with Rank@1 (R@1), Rank@5 (R@5), and mean intersection over union (mIoU). Instead of consolidating all human annotations into one ground truth, we compute the score for a prediction and each human annotation for a particular description/moment pair. To account for outlier annotations, we consider the highest score among sets of annotations A' where A' are the four-choose-three combinations of all four annotations A. Hence, our final score for a prediction P and four human annotations A using metric M is: $score(P, A) = \max_{A' \in \binom{A}{3}} \frac{1}{3} \sum_{a \in A'} M(P, a)$. As not all annotators agree on start and end points it is impossible to achieve 100% on all metrics (c.f., upper bounds in Table 3).

Baseline: Moment Frequency Prior. Though annotators may mark any contiguous set of gifs as a moment, they tend to select short moments toward the beginning of videos. The moment frequency prior selects moments which correspond to gifs most frequently described by annotators.

Baseline: CCA. Canonical correlation analysis (CCA) achieves competitive results for both natural language image [18] and object [27] retrieval tasks. We use the CCA model of [18] and employ the same visual features as the MCN model. We extract language features from our best MCN language encoder for fair comparison.

Baseline: Natural Language Object Retrieval. Natural

language object retrieval models localize objects in a text image. We verify that localizing objects is not sufficient for moment retrieval by running a natural language object retrieval model [14] on videos in our test set. For every tenth frame in a video, we score candidate bounding boxes with the object retrieval model proposed in [14] and compute the score for a frame as the maximum score of all bounding boxes. The score for each candidate moment is the average of scores for frames within the moment. Additionally, we re-train [14] using the same feautures used to train our MCN model; instead of candidate bounding boxes, we provide candidate temporal chunks and train with both appearance and flow input modalities. More details, baselines, and ablations can be found in our appendix.

Implementation Details. DiDeMo videos are split into training (8,395), validation (1,065), and testing (1,004) sets. Videos from a specific Flickr user only appear in one set. All models are implemented in Caffe [15] and have been publicly released [4]. SGD (mini-batch size of 120) is used for optimization and all hyperparamters, such as embedding size (100), margin (0.1), and LSTM hidden state size (1000), are chosen through ablation studies.

5.1. Results

Table 3 compares different variants of our proposed retrieval model to our baselines. Our ablations demonstrate the importance of our temporal context features and the need for both appearance and optical flow features.

Baseline Comparison. Rows 1-7 of Table 3 compare the Moment Context Network (MCN) model to baselines on our test set. Though all baselines we trained (lines 4-6) have similar R@1 and R@5 performance, CCA performs substantially better on the mIoU metric. Scoring video segments based on the scores from a natural language object retrieval model [14] does fairly well, performing similarly to the same model retrained with our features. This suggests that pre-training with a dataset designed for natural language object retrieval and incorporating spatial localization into our model could improve results. We believe that retraining [14] leads to poor results on our dataset because it relies on sentence generation rather than directly retrieving a moment. Additionally, our model does substantially better than the moment frequency prior.

Visual Temporal Context Feature. Rows 9-12 of Table 3 demonstrate the importance of temporal context for moment retrieval. The inclusion of both the global video feature and temporal endpoint feature increase performance considerably. Additionally, we find that combining both appearance and optical flow features is important for best performance.

Qualitative Results. Figure 4 shows moments predicted

---

[4]https://people.eecs.berkeley.edu/~lisa_anne/didemo.html

Query: "first time cat jumps up"

Query: "camera zooms in on group of women"

Query: "both men stop and clasp hands before resuming their demonstration"

Figure 4: Natural language moment retrieval results on DiDeMo. Ground truth moments are outlined in yellow. The Moment Context Network (MCN) localizes diverse descriptions which include temporal indicators, such as "first" (top), and camera words, such as "camera zooms" (middle).

"A ball flies over the athletes."
"A man in a red hat passed a man in a yellow shirt."

Figure 5: MCN correctly retrieves two different moments (light green rectangle on left and light blue rectangle on right). Though our ground truth annotations are five-second segments, we can evaluate with more fine-grained temporal proposals at test time. This gives a better understanding of when moments occur in video (e.g., "A ball flies over the athletes" occurs at the start of the first temporal segment).

by MCN. Our model is capable of localizing a diverse set of moments including moments which require understanding temporal indicators like "first" (Figure 4 top) as well as moments which include camera motion (Figure 4 middle). More qualitative results are in our appendix.

Fine-grained Moment Localization Even though our ground truth moments correspond to five-second chunks, we can evaluate our model on smaller temporal segments at test time to predict moment locations with finer granularity. Instead of extracting features for a five second segment, we evaluate on individual frames extracted at ~ 3 fps. Figure 5 includes an example in which two text queries ("A

ball flies over the athletes" and "A man in a red hat passed a man in a yellow shirt") are correctly localized by our model. The frames which best correspond to "A ball flies over the athletes" occur in the first few seconds of the video and the moment "A man in a red hat passed a men in a yellow shirt" finishes before the end point of the fifth segment. More qualitative results are in our appendix.

Discussion. We introduce the task of localizing moments in video with natural language in a challenging, open-world setting. Our Moment Context Network (MCN) localizes video moments by harnessing local video features, global video features, and temporal endpoint features. To train and evaluate natural language moment localization models, we collect DiDeMo, which consists of over 40,000 pairs of localized moments and referring expressions. Though MCN properly localizes many natural language queries in video, there are still many remaining challenges. For example, modeling complex (temporal) sentence structure is still very challenging (e.g., our model fails to localize "dog stops, then starts rolling around again"). Additionally, DiDeMo has a long-tail distribution with rare activities, nouns, and adjectives. More advanced (temporal) language reasoning and improving generalization to previously unseen vocabulary are two potential future directions.

Acknowledgements

Supplemental

This appendix includes the following material:

1. Qualitative examples illustrating when global video features and tef features improve performance.

2. Qualitative examples contrasting RGB and flow input modalities.

3. Additional qualitative examples using the full Moment Context Network. See `https://www.youtube.com/watch?v=MRO7_4ouNWU` for a video example.

4. Additional baselines.

5. Ablation of inter-intra negative loss.

6. Results when training without a language feature.

7. List of words used to generate numbers in Table 2 of the main paper.

8. Qualitative video retrieval experiment. See `https://www.youtube.com/watch?v=fuz-UBvgapk` for a video example.

9. Discussion on ambiguity of annotations and our metrics.

10. Histrogram showing the moments annotators mark in our dataset.

11. Example video showing our annotation tool (see `https://www.youtube.com/watch?v=vAvT5Amp408` and `https://www.youtube.com/watch?v=9WWgndeEjMU`.

A. Impact of Global Video Features and TEF Features

In the main paper we quantitatively show that global video features and tef features improve model performance. Here, we highlight qualitative examples where the global video features and tef features lead to better localization.

Figure 6 shows examples in which including global context improves performance. Examples like "The car passes the closest to the camera" require context to identify the correct moment. This is sensible as the word "closest" is comparative in nature and determining when the car is closest requires viewing the entire video. Other moments which are correctly localized with context include "we first see the second baby" and "the dog reaches the top of the stairs".

Figure 7 shows examples in which including temporal endpoint features (tef) correctly localizes a video moment. For moments like "we first see the people" the model without tef retrieves a video moment with people, but fails to retrieve the moment when the people first appear. Without

the tef, the model has no indication of *when* a moment occurs in a video. Thus, though the model can identify if there are people in a moment, the model is unable to determine when the people first appear. Likewise, for moments like "train begins to move", the model without tef retrieves a video moment in which the train is moving, but not a moment in which the train begins to move.

B. RGB and Flow Input Modalities

In the main paper, we demonstrate that RGB and optical flow inputs are complementary. Here we show a few examples which illustrate how RGB and flow input modalities complement each other. Figure 8 compares a model trained with RGB input and a model trained with optical flow input (both trained with global video features and tef). We expect the model trained with RGB to accurately localize moments which require understanding the appearance of objects and people in a scene, such as "child jumps into arms of man wearing yellow shirt" (Figure 8 top row). We expect the model trained with flow to better localize moments which require understanding of motion (including camera motion) such as "a dog looks at the camera and jumps at it" and "camera zooms in on a man playing the drums" (Figure 8 row 3 and 4). Frequently, both RGB and optical flow networks can correctly localize a moment (Figure 8 bottom row). However, for best results we take advantage of the complimentary nature of RGB and optical flow input modalities in our fusion model.

C. Qualitative Results for MCN

Figure 9 shows four videos in which we evaluate with fine-grained temporal windows at test time. Observing the plots in Figure 9 provides insight into the exact point at which a moment occurs. For example, our model correctly localizes the phrase "the blue trashcan goes out of view" (Figure 9 bottom right). The finegrained temporal segments that align best with this phrase occur towards the end of the third segment (approximately 14s). Furthermore, Figure 9 provides insight into which parts of the video are most similar to the text query, and which parts are most dissimilar. For example, for the phrase "the blue trashcan goes out of view", there are two peaks; the higher peak occurs when the blue trashcan goes out of view, and the other peak occurs when the blue trashcan comes back into view.

In the main paper, running a natural language object retrieval (NLOR) model on our data is a strong baseline. We expect this model to perform well on examples which require recognizing a specific object such as "a man in a brown shirt runs by the camera" (Figure10 top row), but not as well for queries which require better understanding of action or camera movement such as "man runs towards camera with baby" (row 2 and 4 in Figure 10). Though the

Moment Context Network performs well on DiDeMo, there are a variety of difficult queries it fails to properly localize, such as "Mother holds up the green board for the third time" (Figure 10 last row).

Please see `https://www.youtube.com/watch?v=MRO7_4ouNWU` for examples of moments correctly retrieved by our model.

D. Additional Baselines

In the main paper we compare MCN to the natural language object retrieval model of [14]. Since the publication of [14], better natural language object retrieval models have been proposed (e.g., [13]). We evaluate [13] on our data, in a similar way to how we evaluated [14] on our data in the main paper (Table 3 Row 5 in the main paper). We extract frames at 10 fps on videos in our test set and use [13] to score each bounding box in an image for our description. The score for a frame is the max score of all bounding boxes in the frame, and the score for a moment is the average of all frames in the moment. We expect this model to do well when the moment descriptions can be well localized by localizing specific objects. Surprisingly, even though CMN outperforms [14] for natural language object retrieval, it does worse than [14] on our data (Table 4 row 6). One possible reason is that [13] relies on parsing subject, relationship, and object triplets in sentences. Sentences in DiDeMo may not fit this structure well, leading to a decrease in performance. Additionally, [13] is trained on MSCOCO [6] and [14] is trained on ReferIt [17]. Though MSCOCO is larger than ReferIt, it is possible that the images in ReferIt are more similar to ours and thus [14] transfers better to our task.

Additionally, we train [16], which is designed for natural language image retrieval, using our data. [16] relies on first running a dependency parser to extract sentence fragments linked in a dependency tree (e.g., "black dog", or "run fast"). It scores an image based on how well sentence fragments match a set of proposed bounding boxes. To train this model for our task, we also extract sentence fragments, but then score temporal regions based on how well sentence fragments match a ground truth temporal region. We train on our data (using a late fusion approach to combine RGB and optical flow), and find that this baseline performs similarly to other baselines (Table 4 row 8). In general, we believe our method works better than other baselines because it considers both positive and negative moments when learning to localize video moments and directly optimizes the R@1 metric.

E. Inter-Intra Negative Loss

In Table 4 we compare results when training with only an inter-negative loss, only an intra-negative loss, and our

	Model	Rank@1	Rank@5	mIoU
	Baseline Comparison (Test Set)			
1	Upper Bound	74.75	100.00	96.05
2	Chance	3.75	22.50	22.64
3	Prior (tef)	19.40	66.38	26.65
4	CCA	16.27	41.82	35.73
5	Natural Lang. Obj. Retrieval (SCRC [14])	16.20	43.94	27.18
6	Natural Lang. Obj. Retrieval (CMN [13])	12.59	38.52	22.50
7	Natural Lang. Obj. Retrieval (SCRC [14] re-trained)	15.57	48.32	30.55
8	Image Retrieval (DeFrag [16] re-trained)	10.61	33.00	28.08
9	MCN (ours)	**28.10**	**78.21**	**41.08**
	Ablations (Validation Set)			
10	MCN: Inter-Neg. Loss	25.58	74.13	39.77
11	MCN Intra-Neg. Loss	26.77	78.13	39.83
12	MCN	**27.57**	**79.69**	**41.70**

Table 4: MCN outperformes baselines (rows 1-8) on our test set. We show ablation studies for our inter-intra negative loss in rows 10-12.

proposed inter-intra negative loss. Considering both types of negatives is important for best performance.

F. Importance of Language Feature

Because we ask annotators to mark any interesting moment and describe it, it is possible that annotators mark visually interesting moments which can be localized without text. We thus train a model with our temporal context features but no text query and observe that this model outperforms chance and the moment frequency prior, but does not perform as well as our full model (25.04, 75.23, and 36.12 on R@1, R@5, and mIoU metrics). This indicates that while understanding what constitutes a "describable" moment can be helpful for natural language moment retrieval, natural language is important to achieve best results on DiDeMo. Because the majority of videos include multiple distinct moments (86%), we believe the gap between model trained with and without language will improve with better video-language modelling.

G. Words Used to Construct Table 2

To construct Table 2 in the main paper, we used the following words:

- Camera words: camera, cameras, zoom, zooms, pan, pans, focus, focuses, frame, cameraman

- Temporal words: first, last, after, before, then, second, final, begin, again, return, third, ends

- Spatial words: left, right, top, bottom, background

Additionally, our vocab size is 7,785 words (which is large considering the total number of words in our dataset - 329,274).

H. Video Retrieval Experiment

We used our model to retrieve five moments closest to a specific text query in our shared embedding space from all videos in our test set (Figure 11). We find that retrieved moments are semantically similar to the provided text query. For example, the query "zoom in on baby" returns moments in which the camera zooms in on babies or young children. A similar query, "camera zooms in" returns example moments of the camera zooming, but the videos do not contain babies. Though the query "the white car passes by" does not always return moments with cars, it returns moments which include semantically similar objects (trains, busses and cars).

Please see `https://www.youtube.com/watch?v=fuz-UBvgapk` for an example of video retrieval results.

I. Annotation Ambiguity

Figure 12 shows an example in which the end point for specific moments are ambiguous. For the query "zoom in on man", three annotators mark the fourth segment in which the camera actively zooms in on the man. However, one annotator marks the segment in which the camera zooms in on the man and the following segment when the camera stays zoomed in on the man before zooming out.

This ambiguity informed how we chose our metrics. Based on the annotations for the query "zoom in on man", it is clear that the moment retrieved by our model should include the fourth segment. Though it is less clear if a moment retrieved by our model must include the fifth segment (which was only marked by one annotator to correspond to the phrase "zoom in on man"), it is clear that a model which retrieves both the fourth and fifth segment is more correct than a model which retrieves the third and fourth segment. When we compute a score for a specific example, we choose the maximum score when comparing the model's result to each four-choose-three combinations of human annotations. This results in scores which reflect the intuition outlined above; a model which retrieves only the fourth segment (and therefore agrees with most annotators) will get a higher score than a model which retrieves the fourth and fifth segment (which only agrees with one annotator). Additionally, a model which retrieves the fourth and fifth segment will receive a higher score than a model which retrieves the third and fourth segment.

Note that if two annotators had marked both the fourth and fifth segment, no retrieved moment would perfectly align with any four choose three combination of annotations. Thus, for some examples, it is impossible for any model to achieve a perfect score. In all our qualitative examples where we mark the "ground truth" moment in green, at least three annotators perfectly agree on the start and end point.

J. Distribution of Annotated Moments

Figure 13 shows the distribution of annotated start and end points in DiDeMo. Moments marked by annotators tend to occur at the beginning of the videos and are short. Though a "prior baseline" which retrieves moments which correspond to the most common start and end points in the dataset does much better than chance, our model significantly outperforms a "prior baseline".

The car passes the closest to the camera.

Brown dog runs at the camera.

The dog reaches the top of the stairs.

We first see the second baby.

A girl and a guy hug each other.

Figure 6: Comparison of moments which are correctly retrieved when including global context, but not when only using local video features. Ground truth moments are outlined in green. Global video features improve results for a variety of moments. For moments like "the car passes the closest to the camera", it is not enough to identify a car but to understand when the car is closer to the camera than in any other moment. For moments like "brown dog runs at the camera", the model must not only identify when the brown dog is running, but when it runs towards the camera.

We first see people.

Second child comes running in.

Vehicle is now the furthest away possible.

Train begins to move.

We first see the cross at the front of the room.

Figure 7: Comparison of moments which are correctly retrieved when including the temporal endpoint feature (tef), but not when only using local and global video features. Ground truth moments are outlined in green. For moments like "we first see the people" the model without tef retrieves a video moment with people, but fails to retrieve the moment when the people first appear. Likewise, for moments like "train begins to move", the model without tef retrieves a video moment in which the train is moving, but not a moment in which the train begins to move.

Child jumps into arms of man wearing yellow shirt.

Flow   RGB
                                        Fusion

A white car is visible.

RGB   Flow
Fusion

A dog looks at the camera and jumps at it.

RGB   Flow
                       Fusion

Camera zooms in on a man playing drums.

RGB   Flow
                                 Fusion

Girl waves to the camera.

RGB
Flow
Fusion

Figure 8: Comparison of moments retrieved using different input modalities (ground truth marked in green). For queries like "A white car is visible" which require recognizing an object, a network trained with RGB performs better whereas for queries like "Camera zooms in on a man playing drums" which require understanding movement, a network trained with optical performs better. For some queries, networks trained with either RGB or optical flow retrieve the correct moment.

Figure 9: Comparison of similarity between text queries and finegrained temporal segments. Though ground truth annotations correspond to five second segments, evaluation with more finegrained segments at test time can provide better insight about where a moment occurs within a specific segment and also provide insight into which other parts of a video are similar to a given text query.

A man in a brown shirt runs by the camera.

The camera zooms in on the guitarist.

Pigs run around in a circle before returning to the shade.

Man runs toward the camera with the baby.

Mother holds up the green board for the third time.

Figure 10: We compare our Moment Context Network (MCN) model to a model trained for natural language object retrieval (NLOR). We expect a model trained for natural language object retrieval to perform well when localizing a query relies on locating a specific object (e.g, a man in a brown shirt). However, in general, the MCN model is able to retrieve correct moments more frequently than a model trained for natural language object retrieval. DiDeMo is a difficult dataset and some queries, such as "mother holds up green board for third time" are not correctly localized by the MCN.

Figure 11: We use our model to retrieve the top moments which correspond to a specific query from the entire test set. Though MCN was not trained to retrieve specific moments from a set of different videos, it is able to retrieve semantically meaningful results. Above we show the top five moments retrieved for four separate text queries. A video showing retrieved momenents can be found here: https://www.youtube.com/watch?v=fuz-UBvgapk.

Zoom in on man.

Figure 12: Humans do not always perfectly agree on start and end points for a moment. In the above example we show annotations (denoted as blue lines) from four separate crowd-sourced annotators. Though three annotators agree that the moment corresponds to the fourth segment, a fourth annotator believes the moment corresponds to both the fourth and fifth segment. Our metrics reflect this ambiguity; a model which retrieves only the fourth segment will receive a high score. A model which retrieves both the fourth and fifth segment will receive a lower score, but it will receive a higher score than a model which retrieves the third and fourth segments (which no annotators marked as the correct start and end point).

Figure 13: Distribution of segments marked in DiDeMo. Moments tend to be short and occur towards the beginning of videos.

References

[1] J.-B. Alayrac, P. Bojanowski, N. Agrawal, J. Sivic, I. Laptev, and S. Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *CVPR*, 2016.

[2] S. Bird, E. Klein, and E. Loper. *Natural language processing with Python*. "O'Reilly Media, Inc.", 2009.

[3] O. Boiman and M. Irani. Detecting irregularities in images and in video. *IJCV*, 2007.

[4] P. Bojanowski, R. Lajugie, E. Grave, F. Bach, I. Laptev, J. Ponce, and C. Schmid. Weakly-supervised alignment of video with text. In *ICCV*, 2015.

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

[6] X. Chen, T.-Y. L. Hao Fang, R. Vedantam, S. Gupta, P. Dollr, and C. L. Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arxiv:1504.00325*, 2015.

[7] P. Das, C. Xu, R. F. Doell, and J. J. Corso. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *CVPR*, 2013.

[8] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013.

[9] M. Gygli, H. Grabner, and L. Van Gool. Video summarization by learning submodular mixtures of objectives. In *CVPR*, 2015.

[10] M. Gygli, Y. Song, and L. Cao. Video2gif: Automatic generation of animated gifs from video. In *CVPR*, 2016.

[11] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell. Localizing moments in video with natural language. *arXiv preprint*, 2017.

[12] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 1997.

[13] R. Hu, M. Rohrbach, J. Andreas, T. Darrell, and K. Saenko. Modeling relationships in referential expressions with compositional modular networks. In *CVPR*, 2017.

[14] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. Natural language object retrieval. In *CVPR*, 2016.

[15] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM*, 2014.

[16] A. Karpathy, A. Joulin, and F. F. F. Li. Deep fragment embeddings for bidirectional image sentence mapping. In *NIPS*, 2014.

[17] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014.

[18] B. Klein, G. Lev, G. Sadeh, and L. Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *CVPR*, 2015.

[19] Y. Li, Y. Song, L. Cao, J. Tetreault, L. Goldberg, A. Jaimes, and J. Luo. Tgif: A new dataset and benchmark on animated gif description. *CVPR*, 2016.

[20] D. Lin, S. Fidler, C. Kong, and R. Urtasun. Visual semantic search: Retrieving videos via complex textual queries. In *ICCV*, 2014.

[21] W. Liu, T. Mei, Y. Zhang, C. Che, and J. Luo. Multi-task deep visual-semantic embedding for video thumbnail selection. In *CVPR*, 2015.

[22] J. Mao, J. Huang, A. Toshev, O. Camburu, A. Yuille, and K. Murphy. Generation and comprehension of unambiguous object descriptions. *CVPR*, 2016.

[23] J. Mas and G. Fernandez. Video shot boundary detection based on color histogram. *Notebook Papers TRECVID2003, Gaithersburg, Maryland, NIST*, 2003.

[24] M. Otani, Y. Nakashima, E. Rahtu, J. Heikkilä, and N. Yokoya. Learning joint representations of videos and sentences with web image search. In *ECCV Workshops*, 2016.

[25] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, 2016.

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

[27] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015.

[28] M. Regneri, M. Rohrbach, D. Wetzel, S. Thater, B. Schiele, and M. Pinkal. Grounding action descriptions in videos. In *TACL*, 2013.

[29] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele. Grounding of textual phrases in images by reconstruction. *ECCV*, 2016.

[30] A. Rohrbach, M. Rohrbach, W. Qiu, A. Friedrich, M. Pinkal, and B. Schiele. Coherent multi-sentence video description with variable level of detail. In *GCPR*, 2014.

[31] A. Rohrbach, M. Rohrbach, and B. Schiele. The long-short story of movie description. In *GCPR*, 2015.

[32] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele. A dataset for movie description. In *CVPR*, 2015.

[33] A. Rohrbach, A. Torabi, M. Rohrbach, N. Tandon, C. Pal, H. Larochelle, A. Courville, and B. Schiele. Movie description. *IJCV*, 2017.

[34] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele. Translating video content to natural language descriptions. In *ICCV*, 2013.

[35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

[36] O. Sener, A. R. Zamir, S. Savarese, and A. Saxena. Unsupervised semantic parsing of video collections. In *ICCV*, 2015.

[37] A. Sharghi, B. Gong, and M. Shah. Query-focused extractive video summarization. In *ECCV*, 2016.

[38] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016.

[39] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[40] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng. Grounded compositional semantics for finding and describing images with sentences. *TACL*, 2014.

[41] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes. Tvsum: Summarizing web videos using titles. In *CVPR*, 2015.

[42] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler. Movieqa: Understanding stories in movies through question-answering. In *CVPR*, 2016.

[43] S. Tellex and D. Roy. Towards surveillance video search by natural language query. In *ACM*, 2009.

[44] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. The new data and new challenges in multimedia research. *arXiv preprint arXiv:1503.01817*, 2015.

[45] A. Torabi, C. Pal, H. Larochelle, and A. Courville. Using descriptive video services to create a large data source for video annotation research. *arXiv preprint arXiv:1503.01070*, 2015.

[46] A. Torabi, N. Tandon, and L. Sigal. Learning language-visual embedding for movie understanding with natural-language. *arXiv preprint arXiv:1609.08124*, 2016.

[47] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence – video to text. In *ICCV*, 2015.

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

[49] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: towards good practices for deep action recognition. In *ECCV*, 2016.

[50] J. Xu, T. Mei, T. Yao, and Y. Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016.

[51] R. Xu, C. Xiong, W. Chen, and J. J. Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *AAAI*, 2015.

[52] H. Yang, B. Wang, S. Lin, D. Wipf, M. Guo, and B. Guo. Unsupervised extraction of video highlights via robust recurrent auto-encoders. In *ICCV*, 2015.

[53] T. Yao, T. Mei, and Y. Rui. Highlight detection with pairwise deep ranking for first-person video summarization. In *CVPR*, 2016.

[54] S. Yeung, A. Fathi, and L. Fei-Fei. Videoset: Video summary evaluation through text. In *CVPR Workshops*, 2014.

[55] H. Yu and J. M. Siskind. Grounded language learning from video described with sentences. In *ACL*, 2013.

[56] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *CVPR*, 2015.