

# Dense-Captioning Events in Videos

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, Juan Carlos Nieves  
Stanford University

{ranjaykrishna, kenjihata, fren, feifeili, jniebles}@cs.stanford.edu

## Abstract

Most natural videos contain numerous events. For example, in a video of a “man playing a piano”, the video might also contain “another man dancing” or “a crowd clapping”. We introduce the task of **dense-captioning events**, which involves both **detecting and describing events** in a video. We propose a new model that is able to **identify all events in a single pass of the video** while simultaneously describing the detected events with natural language. Our model introduces a variant of an **existing proposal module** that is **designed to capture both short as well as long events that span minutes**. To capture the **dependencies** between the events in a video, our model introduces a new captioning module that uses contextual information from past and future events to **jointly describe all events**. We also introduce **ActivityNet Captions**, a large-scale benchmark for **dense-captioning events**. ActivityNet Captions contains 20k videos amounting to 849 video hours with 100k total descriptions, each with its unique start and end time. Finally, we **report performances of our model for dense-captioning events, video retrieval and localization**.

Dense Captioning Model , ActivityNet Captions Set

## 1. Introduction

With the introduction of large scale activity datasets [26, 21, 15, 4], it has become possible to categorize videos into a discrete set of action categories [32, 13, 11, 52, 46]. For example, in Figure 1, such models would output labels like *playing piano* or *dancing*. While the success of these methods is encouraging, **they all share one key limitation: detail**. To elevate the lack of detail from existing action detection models, subsequent work has explored explaining video semantics using sentence descriptions [34, 38, 33, 50, 49]. For example, in Figure 1, such models would likely concentrate on *an elderly man playing the piano in front of a crowd*. While this caption provides us more details about who is playing the piano and mentions an audience, **it fails to recognize and articulate all the other events in the video**. For example, at some point in the video, *a woman starts singing along with the pianist* and then later *another man starts*

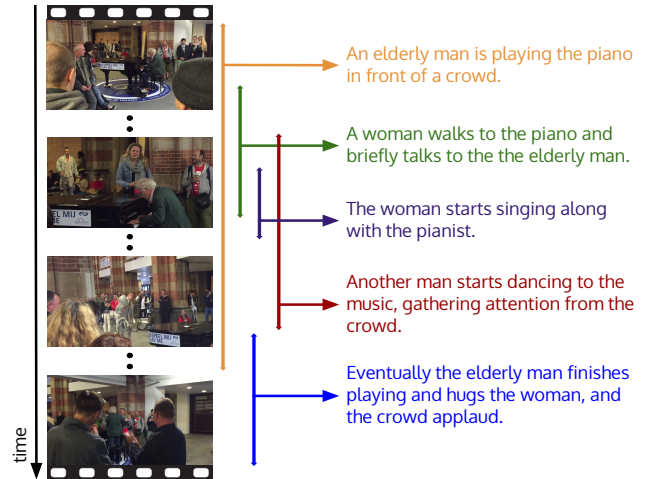


Figure 1: Dense-captioning events in a video involves detecting multiple events that occur in a video and describing each event using natural language. These events are temporally localized in the video with independent start and end times, resulting in some events that might also occur concurrently and overlap in time.

*dancing to the music*. In order to **identify all the events in a video and describe them in natural language**, we introduce the task of **dense-captioning events**, which requires a model to generate a set of descriptions for multiple events occurring in the video and localize them in time.

**Dense-captioning events is analogous to dense-image-captioning [18]**; it describes videos and localizes events in time whereas dense-image-captioning describes and localizes regions in space. However, we observe that dense-captioning events comes **with its own set of challenges distinct from the image case**. One observation is that events in videos can **range across multiple time scales** and can even **overlap**. While *piano recitals* might last for the entire duration of a long video, *the applause* takes place in a couple of seconds. To **capture all such events**, we need to **design ways of encoding short as well as long sequences** of video frames **to propose events**. Past captioning works have circumvented this problem by encoding the entire video se-

quence by mean-pooling [50] or by using a recurrent neural network (RNN) [49]. While this works well for short clips, encoding long video sequences that span minutes leads to vanishing gradients, preventing successful training. To overcome this limitation, we **extend recent work on generating action proposals [10] to multi-scale detection of events**. Also, our proposal module processes each video in a forward pass, allowing us to detect events as they occur.

Another key observation is that the events in a given video are usually related to one another. In Figure 1, *the crowd applauds because a man was playing the piano*. Therefore, our model must be **able to use context from surrounding events to caption each event**. A recent paper has attempted to describe videos **with multiple sentences [64]**. However, **their model generates sentences for instructional “cooking” videos where the events occur sequentially and highly correlated to the objects in the video [37]**. We show that their model does not generalize to “open” domain videos where events are action oriented and can even overlap. We introduce a **captioning module that utilizes the context** from all the **events from our proposal module** to generate each sentence. In addition, we show a variant of our captioning module that can operate on streaming videos by attending over only the past events. Our full model attends over **both past as well as future events and demonstrates the importance of using context**.

To evaluate our model and benchmark progress in dense-captioning events, we introduce the **ActivityNet Captions dataset<sup>1</sup>**. ActivityNet Captions contains 20k videos taken from ActivityNet [4], where each video is annotated **with a series of temporally localized descriptions** (Figure 1). To showcase **long term event detection**, our dataset contains videos **as long as 10 minutes**, with each video annotated **with on average 3.65 sentences**. The descriptions refer to events that might be simultaneously occurring, causing the video segments to overlap. We ensure that **each description in a given video is unique and refers to only one segment**. While our videos are centered around **human** activities, the descriptions may also refer to **non-human** events such as: *two hours later, the mixture becomes a delicious cake to eat*. We collect our descriptions using crowdsourcing find that there is high agreement in the temporal event segments, which is in line with research suggesting that brain activity is naturally structured into semantically meaningful events [2].

With ActivityNet Captions, we are able to provide the first results for the task of dense-captioning events. Together with our online proposal module and our online captioning module, **we show that we can detect and describe**

**events in long or even streaming videos**. We demonstrate that we are able to detect events found in **short** clips as well as in **long** video sequences. Furthermore, we show that utilizing **context** from other events in the video improves dense-captioning events. Finally, we demonstrate how ActivityNet Captions can be used to study video retrieval as well as event localization.

## 2. Related work

**Dense-captioning events bridges two separate bodies of work: temporal action proposals and video captioning**. First, we review related work on action recognition, action detection and temporal proposals. Next, we survey how video captioning started from video retrieval and video summarization, leading to single-sentence captioning work. Finally, we **contrast our work with recent work in captioning images and videos with multiple sentences**.

Early work in **activity recognition** involved using hidden **Markov models** to learn latent action states [58], followed by discriminative **SVM** models that used key poses and action grammars [31, 48, 35]. Similar works have used **hand-crafted features** [40] or **object-centric features** [30] to recognize actions in fixed camera settings. **More recent works have used dense trajectories [51] or deep learning features [19] to study actions**. While our work is similar to these methods, we focus on **describing such events with natural language instead of a fixed label set**.

To enable action localization, **temporal action proposal methods started from traditional sliding window approaches [9] and later started building models to propose a handful of possible action segments [10, 5]**. These proposal methods have used dictionary learning [5] or RNN architectures [10] to find possible segments of interest. However, such methods **required each video frame to be processed once for every sliding window**. DAPs introduced a framework to allow proposing overlapping segments using a sliding window. We modify this framework **by removing the sliding windows and outputting proposals at every time step in a single pass of the video**. We further extend this model and enable it to detect long events by implementing a multi-scale version of **DAPs**, where we sample frames at longer strides.

Orthogonal to work studying proposals, early approaches that connected video with language studied the task of **video retrieval with natural language**. They worked on **generating a common embedding space between language and videos [33, 57]**. Similar to these, we evaluate how well existing models perform on our dataset. Additionally, we introduce the task of localizing a given sentence given a video frame, allowing us to now also evaluate whether our models are able to locate specified events.

In an effort to start describing videos, methods in **video summarization** aimed to congregate segments of videos

<sup>1</sup>The dataset is available at <http://cs.stanford.edu/people/ranjaykrishna/densevid/>. For a detailed analysis of our dataset, please see our supplementary material.

that include important or interesting visual information [62, 59, 16, 3]. These methods attempted to use low level features such as color and motion or attempted to model objects [65] and their relationships [53, 14] to **select key segments**. Meanwhile, others have utilized text inputs from user studies to guide the selection process [44, 27]. While these summaries **provide a means of finding important segments**, these methods are limited by **small vocabularies** and do not evaluate how well we can explain visual events [63].

After these summarization works, **early attempts at video captioning** [50] simply **mean-pooled** video frame features and used a pipeline inspired by the success of image captioning [20]. However, this approach only works **for short video clips with only one major event**. To avoid this issue, others have proposed **either a recurrent encoder** [8, 49, 54] **or an attention mechanism** [61]. To capture more detail in videos, a new paper has recommended describing videos **with paragraphs** (a list of sentences) using **a hierarchical RNN** [29] where the top level network generates a series of hidden vectors that are used to initialize low level RNNs that generate each individual sentence [64]. **While our paper is most similar to this work**, we address two important **missing factors**. First, the sentences that their model generates **refer to different events in the video but are not localized in time**. Second, they use the TACoS-MultiLevel [37], which **contains less than 200 videos and is constrained to “cooking” videos** and only contain non-overlapping **sequential events**. We address these issues by introducing the **ActivityNet Captions** dataset which contains overlapping events and by introducing our captioning module that uses **temporal context** to capture the interdependency between all the events in a video.

Finally, we build upon the recent work on **dense-image-captioning** [18], which generates a set of **localized descriptions** for an image. Further work for this task has used **spatial context to improve captioning** [60, 56]. Inspired by this work, and by recent literature on using spatial attention to improve human tracking [1], we design our captioning module to **incorporate temporal context** (analogous to spatial context except in time) **by attending over the other events in the video**.

### 3. Dense-captioning events model

**Overview.** Our goal is to **design an architecture** that jointly **localizes temporal proposals of interest and then describes each with natural language**. The **two main challenges** we face are to develop a method that can (1) **detect multiple events** in short as well as long video sequences and (2) **utilize the context** from past, concurrent and future events to **generate descriptions of each one**. Our **proposed architecture** (Figure 2) **draws on architectural elements present in recent work on action proposal [10] and social human tracking [1] to tackle both these challenges**.

Formally, the input to our system is a **sequence of video frames**  $v = \{v_t\}$  where  $t \in 0, \dots, T - 1$  indexes the frames in temporal order. Our output is a set of sentences  $s_i \in \mathcal{S}$  where  $s_i = (t_i^{\text{start}}, t_i^{\text{end}}, \{v_j\})$  consists of the start and end times for each sentence which is defined by a set of words  $v_j \in V$  with differing lengths for each sentence and  $V$  is our vocabulary set.

Our model first **sends the video frames through a proposal module that generates a set of proposals**:

$$P = \{(t_i^{\text{start}}, t_i^{\text{end}}, \text{score}_i, h_i)\} \quad (1)$$

Proposal : 起始时间+置信分数+hidden representation

All the proposals with a  $\text{score}_i$  higher than a **threshold** are forwarded to our language model that uses context from the other proposals while captioning each event. The hidden representation  $h_i$  of the event proposal module is used as inputs to the captioning module, which then outputs descriptions for each event, while utilizing the context from the other events.

#### 3.1. Event proposal module

The proposal module in Figure 2 **tackles the challenge of detecting events in short as well as long video sequences**, while preventing the dense application of our language model over sliding windows during inference. **Prior work** usually **pools video features globally into a fixed sized vector** [8, 49, 54], which is sufficient for representing short video clips but is **unable to detect multiple events** in long videos. Additionally, we would like to detect events in a single pass of the video so that the gains over a simple temporal sliding window are significant. To tackle this challenge, **we design an event proposal module to be a variant of DAPs [10] that can detect longer events**.

**Input.** Our **proposal module receives** a series of **features capturing semantic information from the video frames**. Concretely, the input to our proposal module is a **sequence of features**:  $\{f_t = F(v_t : v_{t+\delta})\}$  where  $\delta$  is the time resolution of each feature  $f_t$ . In our paper,  $F$  **extracts C3D features [17] where  $\delta = 16$  frames**. The output of  $F$  is a tensor of size  $N \times D$  where  $D = 500$  dimensional features and  $N = T/\delta$  discretizes the video frames.

每16帧丢入C3D做一个feature encoding。

N个feature组成序列。

**DAPs.** Next, **we feed these features into a variant of DAPs [10] where we sample the videos features at different strides** (1, 2, 4 and 8 for our experiments) and feed them into a proposal long short-term memory (LSTM) unit. **The longer strides are able to capture longer events**. The LSTM accumulates evidence across time as the video features progress. We do not modify the training of DAPs and **only change the model at inference time** by outputting  $K$  proposals at every time step, each proposing an event with offsets. So, the LSTM is capable of generating proposals at different **overlapping time intervals** and we only need to iterate over the video once, since all the strides can be computed in parallel. Whenever the proposal LSTM detects an

feature sequence并不是全都过一遍LSTM。而是按照一定的间隔均匀采样丢入LSTM。间隔取1, 2, 4, 8。更长的stride可以保证对更长的event做捕捉。(信息损失小。)

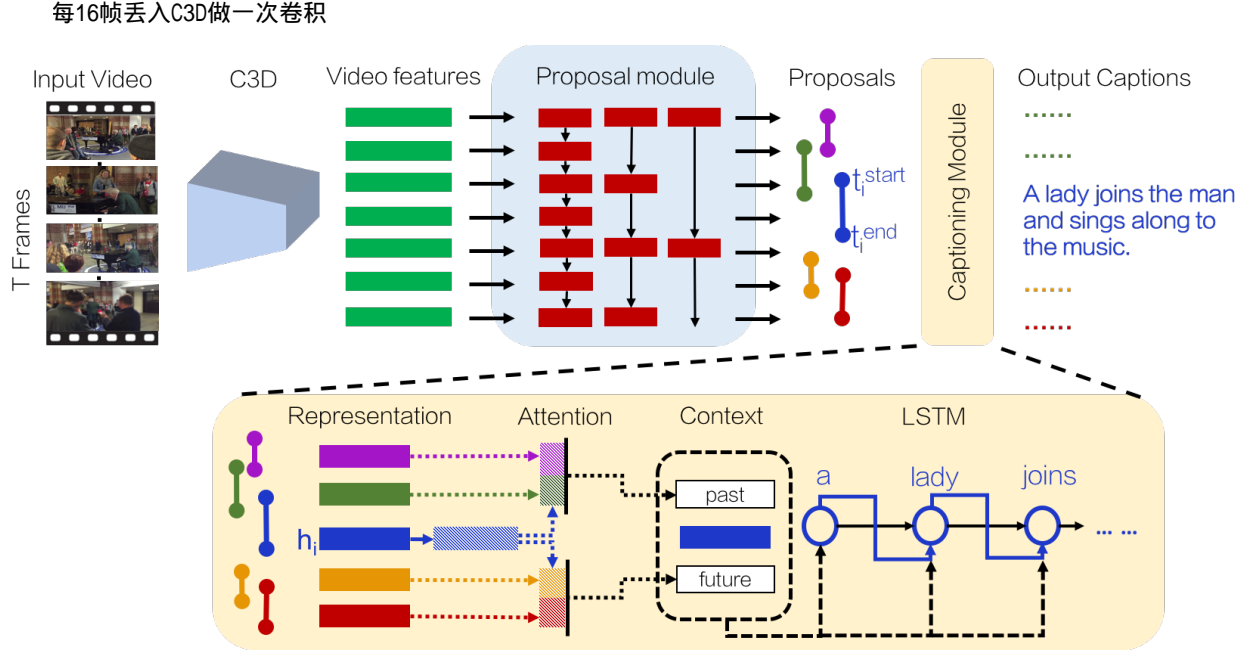


Figure 2: Complete pipeline for dense-captioning events in videos with descriptions. We first extract C3D features from the input video. These features are fed into our proposal module at varying stride to predict both short as well as long events. Each proposal, which consists of a unique start and end time and a hidden representation, is then used as input into the captioning module. Finally, this captioning model leverages context from neighboring events to generate each event description.

event, we use the hidden state of the LSTM at that time step as a feature representation of the visual event. Note that the proposal model can output proposals for events that can be overlapping. While traditional DAPs uses non-maximum suppression to eliminate overlapping outputs, we keep them separately and treat them as individual events.

### 3.2. Captioning module with context

Once we have the event proposals, the next stage of our pipeline is responsible for describing each event. A naive captioning approach could treat each description individually and use a captioning LSTM network to describe each one. However, most events in a video are correlated and can even cause one another. For example, we saw in Figure 1 that the man playing the piano caused the other person to start dancing. We also saw that after the man finished playing the piano, the audience applauded. To capture such correlations, we design our captioning module to incorporate the context from its neighboring events. Inspired by recent work [1] on human tracking that utilizes spatial context between neighboring tracks, we develop an analogous model that captures temporal context in videos by grouping together events in time instead of tracks in space.

**Incorporating context.** To capture the context from all other neighboring events, we categorize all events into two buckets relative to a reference event. These two context buckets capture events that have already occurred (past), and events that take place after this event has finished (fu-

ture). Concurrent events are split into one of the two buckets: past if it end early and future otherwise. For a given video event from the proposal module, with hidden representation  $h_i$  and start and end times of  $[t_i^{\text{start}}, t_i^{\text{end}}]$ , we calculate the past and future context representations as follows:

$$h_i^{\text{past}} = \frac{1}{Z^{\text{past}}} \sum_{j \neq i} \mathbb{1}[t_j^{\text{end}} < t_i^{\text{end}}] w_j h_j \quad (2)$$

$$h_i^{\text{future}} = \frac{1}{Z^{\text{future}}} \sum_{j \neq i} \mathbb{1}[t_j^{\text{end}} \geq t_i^{\text{end}}] w_j h_j \quad (3)$$

where  $h_j$  is the hidden representation of the other proposed events in the video.  $w_j$  is the weight used to determine how relevant event  $j$  is to event  $i$ .  $Z$  is the normalization that is calculated as  $Z^{\text{past}} = \sum_{j \neq i} \mathbb{1}[t_j^{\text{end}} < t_i^{\text{end}}]$ . We calculate  $w_j$  as follows:

$$a_i = w_a h_i + b_a \quad (4)$$

$$w_j = a_i h_j \quad (5)$$

where  $a_i$  is the attention vector calculated from the learnt weights  $w_a$  and bias  $b_a$ . We use the dot product of  $a_i$  and  $h_j$  to calculate  $w_j$ . The concatenation of  $(h_i^{\text{past}}, h_i, h_i^{\text{future}})$  is then fed as the input to the captioning LSTM that describes the event. With the help of the context, each LSTM also has knowledge about events that have happened or will happen and can tune its captions accordingly.



**Language modeling.** Each language LSTM is initialized to have 2 layers with 512 dimensional hidden representation. We randomly initialize all the word vector embeddings from a Gaussian with standard deviation of 0.01. We sample predictions from the model using beam search of size 5.

### 3.3. Implementation details.

**Loss function.** We use two separate losses to train both our proposal model ( $\mathcal{L}_{\text{prop}}$ ) and our captioning model ( $\mathcal{L}_{\text{cap}}$ ). Our proposal models predicts confidences ranging between 0 and 1 for varying proposal lengths. We use a weighted cross-entropy term to evaluate each proposal confidence.

We only pass to the language model proposals that have a high IoU with ground truth proposals. Similar to previous work on language modeling [22, 20], we use a cross-entropy loss across all words in every sentence. We normalize the loss by the batch-size and sequence length in the language model. We weight the contribution of the captioning loss with  $\lambda_1 = 1.0$  and the proposal loss with  $\lambda_2 = 0.1$ :

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{cap}} + \lambda_2 \mathcal{L}_{\text{prop}} \quad (6)$$

**Training and optimization.** We train our full dense-captioning model by alternating between training the language model and the proposal module every 500 iterations. We first train the captioning module by masking all neighboring events for 10 epochs before adding in the context features. We initialize all weights using a Gaussian with standard deviation of 0.01. We use stochastic gradient descent with momentum 0.9 to train. We use an initial learning rate of  $1 \times 10^{-2}$  for the language model and  $1 \times 10^{-3}$  for the proposal module. For efficiency, we do not finetune the C3D feature extraction.

Our training batch-size is set to 1. We cap all sentences to be a maximum sentence length of 30 words and implement all our code in PyTorch 0.1.10. One mini-batch runs in approximately 15.84 ms on a Titan X GPU and it takes 2 days for the model to converge.

## 4. ActivityNet Captions dataset

The ActivityNet Captions dataset connects videos to a series of temporally annotated sentences. Each sentence covers an unique segment of the video, describing an event that occurs. These events may occur over very long or short periods of time and are not limited in any capacity, allowing them to co-occur. We will now present an overview of the dataset and also provide a detailed analysis and comparison with other datasets in our supplementary material.

### 4.1. Dataset statistics

On average, each of the 20k videos in ActivityNet Captions contains 3.65 temporally localized sentences, resulting in a total of 100k sentences. We find that the number of

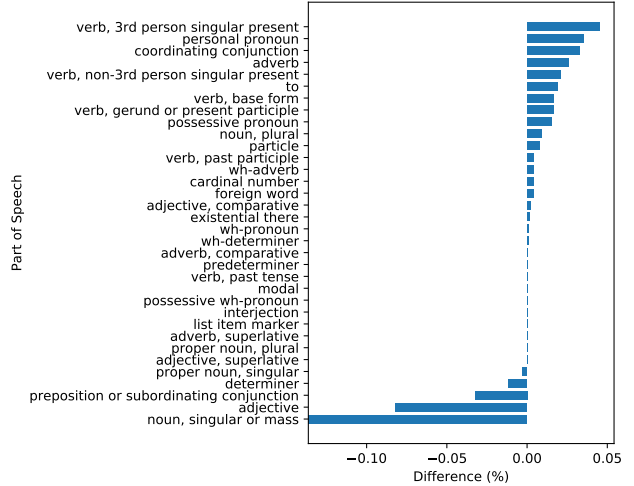


Figure 3: The parts of speech distribution of ActivityNet Captions compared with Visual Genome, a dataset with multiple sentence annotations per image. There are many more verbs and pronouns represented in ActivityNet Captions, as the descriptions often focus on actions.

sentences per video follows a relatively normal distribution. Furthermore, as the video duration increases, the number of sentences also increases. Each sentence has an average length of 13.48 words, which is also normally distributed.

On average, each sentence describes 36 seconds and 31% of their respective videos. However, the entire paragraph for each video on average describes 94.6% of the entire video, demonstrating that each paragraph annotation still covers all major actions within the video. Furthermore, we found that 10% of the temporal descriptions overlap, showing that the events cover simultaneous events.

Finally, our analysis on the sentences themselves indicate that ActivityNet Captions focuses on verbs and actions. In Figure 3, we compare against Visual Genome [23], the image dataset with most number of image descriptions (4.5 million). With the percentage of verbs comprising ActivityNet Captions being significantly more, we find that ActivityNet Captions shifts sentence descriptions from being object-centric in images to action-centric in videos. Furthermore, as there exists a greater percentage of pronouns in ActivityNet Captions, we find that the sentence labels will more often refer to entities found in prior sentences.

### 4.2. Temporal agreement amongst annotators

To verify that ActivityNet Captions's captions mark semantically meaningful events [2], we collected two distinct, temporally annotated paragraphs from different workers for each of the 4926 validation and 5044 test videos. Each pair of annotations was then tested to see how well they temporally corresponded to each other. We found that, on aver-

	with GT proposals						with learnt proposals					
	B@1	B@2	B@3	B@4	M	C	B@1	B@2	B@3	B@4	M	C
LSTM-YT [49]	18.22	7.43	3.24	1.24	6.56	14.86	-	-	-	-	-	-
S2VT [50]	20.35	8.99	4.60	2.62	7.85	20.97	-	-	-	-	-	-
H-RNN [64]	19.46	8.78	4.34	2.53	8.02	20.18	-	-	-	-	-	-
no context (ours)	20.35	8.99	4.60	2.62	7.85	20.97	12.23	3.48	2.10	0.88	3.76	12.34
online—attn (ours)	21.92	9.88	5.21	3.06	8.50	22.19	15.20	5.43	2.52	1.34	4.18	14.20
online (ours)	22.10	10.02	5.66	3.10	8.88	22.94	17.10	7.34	3.23	1.89	4.38	15.30
full—attn (ours)	26.34	13.12	6.78	3.87	9.36	24.24	15.43	5.63	2.74	1.72	4.42	15.29
full (ours)	<b>26.45</b>	<b>13.48</b>	<b>7.12</b>	<b>3.98</b>	<b>9.46</b>	<b>24.56</b>	<b>17.95</b>	<b>7.69</b>	<b>3.86</b>	<b>2.20</b>	<b>4.82</b>	<b>17.29</b>

Table 1: We report Bleu (B), METEOR (M) and CIDEr (C) captioning scores for the task of dense-captioning events. On the left, we report performances of just our captioning module with ground truth proposals. On the right, we report the combined performances of our complete model, with proposals predicted from our proposal module. Since prior work has focused only on describing entire videos and not also detecting a series of events, we only compare existing video captioning models using ground truth proposals.

age, each sentence description had an tIoU of 70.2% with the maximal overlapping combination of sentences from the other paragraph. Since these results agree with prior work [2], we found that workers generally agree with each other when annotating temporal boundaries of video events.

## 5. Experiments

We evaluate our model by detecting multiple events in videos and describing them. We refer to this task as dense-captioning events (Section 5.1). We test our model on ActivityNet Captions, which was built specifically for this task.

Next, we provide baseline results on two additional tasks that are possible with our model. The first of these tasks is localization (Section 5.2), which tests our proposal model’s capability to adequately localize all the events for a given video. The second task is retrieval (Section 5.3), which tests a variant of our model’s ability to recover the correct set of sentences given the video or vice versa. Both these tasks are designed to test the event proposal module (localization) and the captioning module (retrieval) individually.

### 5.1. Dense-captioning events

To dense-caption events, our model is given an input video and is tasked with detecting individual events and describing each one with natural language.

**Evaluation metrics.** Inspired by the dense-image-captioning [18] metric, we use a similar metric to measure the joint ability of our model to both localize and caption events. This metric computes the average precision across tIoU thresholds of 0.3, 0.5, 0.7 when captioning the top 1000 proposals. We measure precision of our captions using traditional evaluation metrics: Bleu, METEOR and CIDEr.

To isolate the performance of language in the predicted captions without localization, we also use ground truth locations across each test image and evaluate predicted captions.

	B@1	B@2	B@3	B@4	M	C
<b>no context</b>						
1 <sup>st</sup> sen.	23.60	12.19	7.11	4.51	9.34	<b>31.56</b>
2 <sup>nd</sup> sen.	19.74	8.17	3.76	1.87	7.79	19.37
3 <sup>rd</sup> sen.	18.89	7.51	3.43	1.87	7.31	19.36
<b>online</b>						
1 <sup>st</sup> sen.	24.93	12.38	7.45	4.77	8.10	30.92
2 <sup>nd</sup> sen.	19.96	8.66	4.01	1.93	7.88	19.17
3 <sup>rd</sup> sen.	19.22	7.72	3.56	<b>1.89</b>	7.41	19.36
<b>full</b>						
1 <sup>st</sup> sen.	<b>26.33</b>	<b>13.98</b>	<b>8.45</b>	<b>5.52</b>	<b>10.03</b>	29.92
2 <sup>nd</sup> sen.	<b>21.46</b>	<b>9.06</b>	<b>4.40</b>	<b>2.33</b>	<b>8.28</b>	<b>20.17</b>
3 <sup>rd</sup> sen.	<b>19.82</b>	<b>7.93</b>	<b>3.63</b>	1.83	<b>7.81</b>	<b>20.01</b>

Table 2: We report the effects of context on captioning the 1<sup>st</sup>, 2<sup>nd</sup> and 3<sup>rd</sup> events in a video. We see that performance increases with the addition of past context in the online model and with future context in full model.

**Baseline models.** Since all the previous models proposed so far have focused on the task of describing entire videos and not detecting a series of events, we only compare existing video captioning models using ground truth proposals. Specifically, we compare our work with *LSTM-YT* [49], *S2VT* [50] and *H-RNN* [64]. *LSTM-YT* pools together video features to describe videos while *S2VT* [50] encodes a video using an RNN. *H-RNN* [64] generates paragraphs by using one RNN to caption individual sentences while the second RNN is used to sequentially initialize the hidden state for the next sentence generation. Our model can be thought of as a generalization of the *H-RNN* model as it uses context, not just from the previous sentence but from surrounding events in the video. Additionally, our method treats context, not as features from object detectors but encodes it from unique parts of the proposal module.

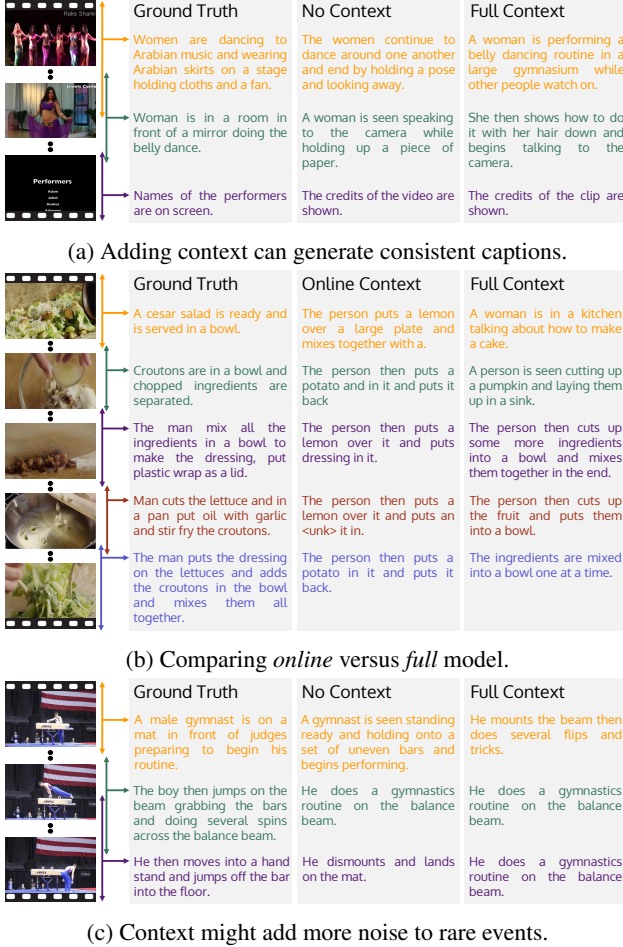


Figure 4: Qualitative dense-captioning captions generated using our model. We show captions with the highest overlap with ground truth captions.

**Variants of our model.** Additionally, we compare different variants of our model. Our *no context* model is our implementation of *S2VT*. The *full* model is our complete model described in Section 3. The *online* model is a version of our full model that uses context only from past events and not from future events. This version of our model can be used to caption long streams of video in a single pass. The *full-attn* and *online-attn* models use mean pooling instead of attention to concatenate features, i.e. it sets  $w_j = 1$  in Equation 5.

**Captioning results.** Since all the previous work has focused on captioning complete videos, We find that *LSTM-YT* performs much worse than other models as it tries to encode long sequences of video by mean pooling their features (Table 1). *H-RNN* performs slightly better but attends over object level features to generate sentence, which causes it to only slightly outperform *LSTM-YT* since we demonstrated earlier that the captions in our dataset are not object centric

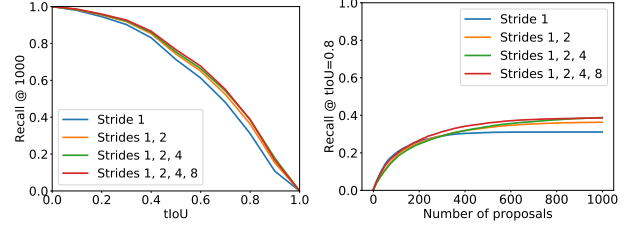


Figure 5: Evaluating our proposal module, we find that sampling videos at varying strides does in fact improve the module’s ability to localize events, specially longer events.

but action centric instead. *S2VT* and our *no context* model performs better than the previous baselines with a CIDEr score of 20.97 as it uses an RNN to encode the video features. We see an improvement in performance to 22.19 and 22.94 when we incorporate context from past events into our *online-attn* and *online* models. Finally, we also considering events that will happen in the future, we see further improvements to 24.24 and 24.56 for the *full-attn* and *full* models. Note that while the improvements from using attention is not too large, we see greater improvements amongst videos with more events, suggesting that attention is useful for longer videos.

**Sentence order.** To further benchmark the improvements calculated from utilizing past and future context, we report results using ground truth proposals for the first three sentences in each video (Table 2). While there are videos with more than three sentences, we report results only for the first three because almost all the videos in the dataset contains at least three sentences. We notice that the *online* and *full* context models see most of their improvements from subsequent sentences, i.e. not the first sentence. In fact, we notice that after adding context, the CIDEr score for the *online* and *full* models tend to decrease for the 1<sup>st</sup> sentence.

**Results for dense-captioning events.** When using proposals instead of ground truth events (Table 1), we see a similar trend where adding more context improves captioning. However, we also see that the improvements from attention are more pronounced since there are many events that the model has to caption. Attention allows the model to adequately focus in on select other events that are relevant to the current event. We show examples qualitative results from the variants of our models in Figure 4. In (a), we see that the last caption in the *no context* model drifts off topic while the *full* model utilizes context to generate more reasonable context. In (c), we see that our *full* context model is able to use the knowledge that the vegetables are later *mixed in the bowl* to also mention *the bowl* in the third and fourth sentences, propagating context back through to past events. However, context is not always successful at generating better captions. In (c), when the proposed segments

	Video retrieval				Paragraph retrieval			
	R@1	R@5	R@50	Med. rank	R@1	R@5	R@50	Med. rank
LSTM-YT [49]	0.00	0.04	0.24	102	0.00	0.07	0.38	98
no context [50]	0.05	0.14	0.32	78	0.07	0.18	0.45	56
online (ours)	0.10	<b>0.32</b>	0.60	36	0.17	0.34	0.70	33
full (ours)	<b>0.14</b>	<b>0.32</b>	<b>0.65</b>	<b>34</b>	<b>0.18</b>	<b>0.36</b>	<b>0.74</b>	<b>32</b>

Table 3: Results for video and paragraph retrieval. We see that the utilization of context to encode video events help us improve retrieval.  $R@k$  measures the recall at varying thresholds  $k$  and med. rank measures the median rank the retrieval.

have a high overlap, our model fails to distinguish between the two events, causing it to repeat captions.

## 5.2. Event localization

One of the main goals of this paper is to develop models that can locate any given event within a video. Therefore, we test how well our model can predict the temporal location of events within the corresponding video, in isolation of the captioning module. Recall that our variant of the proposal module uses proposes videos at different strides. Specifically, we test with strides of 1, 2, 4 and 8. Each stride can be computed in parallel, allowing the proposal to run in a single pass.

**Setup.** We evaluate our proposal module using recall (like previous work [10]) against (1) the number of proposals and (2) the IoU with ground truth events. Specifically, we are testing whether, the use of different strides does in fact improve event localization.

**Results.** Figure 5 shows the recall of predicted localizations that overlap with ground truth over a range of IoU’s from 0.0 to 1.0 and number of proposals ranging till 1000. We find that using more strides improves recall across all values of IoU’s with diminishing returns. We also observe that when proposing only a few proposals, the model with stride 1 performs better than any of the multi-stride versions. This occurs because there are more training examples for smaller strides as these models have more video frames to iterate over, allowing them to be more accurate. So, when predicting only a few proposals, the model with stride 1 localizes the most correct events. However, as we increase the number of proposals, we find that the proposal network with only a stride of 1 plateaus around a recall of 0.3, while our multi-scale models perform better.

## 5.3. Video and paragraph retrieval

While we introduce dense-captioning events, a new task to study video understanding, we also evaluate our intuition to use context on a more traditional task: video retrieval.

**Setup.** In video retrieval, we are given a set of sentences that describe different parts of a video and are asked to retrieve the correct video from the test set of all videos. Our retrieval model is a slight variant on our dense-captioning

model where we encode all the sentences using our captioning module and then combine the context together for each sentence and match each sentence to multiple proposals from a video. We assume that we have ground truth proposals for each video and encode each proposal using the LSTM from our proposal model. We train our model using a max-margin loss that attempts to align the correct sentence encoding to its corresponding video proposal encoding. We also report how this model performs if the task is reversed, where we are given a video as input and are asked to retrieve the correct paragraph from the complete set of paragraphs in the test set.

**Results.** We report our results in Table 3. We evaluate retrieval using recall at various thresholds and the median rank. We use the same baseline models as our previous tasks. We find that models that use RNNs (*no context*) to encode the video proposals perform better than max pooling video features (LSTM-YT). We also see a direct increase in performance when context is used. Unlike dense-captioning, we do not see a marked increase in performance when we include context from future events as well. We find that our online models performs almost at par with our full model.

## 6. Conclusion

We introduced the task of dense-captioning events and identified two challenges: (1) events can occur within a second or last up to minutes, and (2) events in a video are related to one another. To tackle both these challenges, we proposed a model that combines a new variant of an existing proposal module with a new captioning module. The proposal module samples video frames at different strides and gathers evidence to propose events at different time scales in one pass of the video. The captioning module attends over the neighboring events, utilizing their context to improve the generation of captions. We compare variants of our model and demonstrate that context does indeed improve captioning. We further show how the captioning model uses context to improve video retrieval and how our proposal model uses the different strides to improve event localization. Finally, this paper also releases a new dataset for dense-captioning events: ActivityNet Captions.



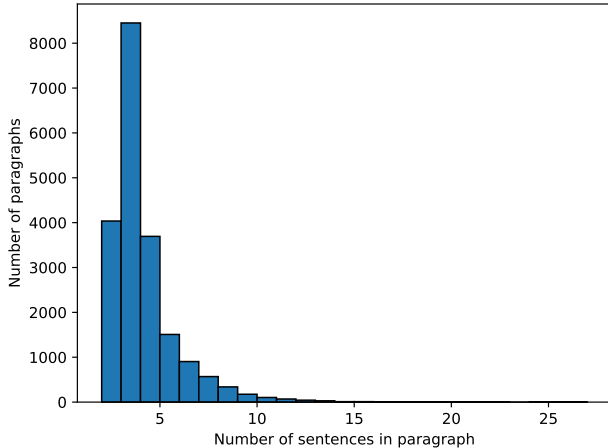


Figure 6: The number of sentences within paragraphs is normally distributed, with on average 3.65 sentences per paragraph.

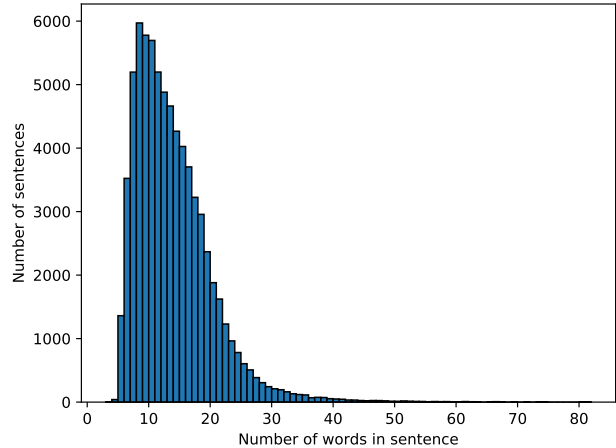


Figure 7: The number of words per sentence within paragraphs is normally distributed, with on average 13.48 words per sentence.

## 7. Supplementary material

In the supplementary material, we compare and contrast our dataset with other datasets and provide additional details about our dataset. We include screenshots of our collection interface with detailed instructions. We also provide additional details about the workers who completed our tasks.

### 7.1. Comparison to other datasets.

Curation and open distribution is closely correlated with progress in the field of video understanding (Table 4). The KTH dataset [42] pioneered the field by studying human actions with a black background. Since then, datasets like UCF101 [45], Sports 1M [21], Thumos 15 [15] have focused on studying actions in sports related internet videos while HMDB 51 [25] and Hollywood 2 [28] introduced a dataset of movie clips. Recently, ActivityNet [4] and Charades [43] broadened the domain of activities captured by these datasets by including a large set of human activities. In an effort to map video semantics with language, MPII MD [39] and M-VAD [47] released short movie clips with descriptions. In an effort to capture longer events, MSR-VTT [55], MSVD [6] and YouCook [7] collected a dataset with slightly longer length, at the cost of a few descriptions than previous datasets. To further improve video annotations, KITTI [12] and TACoS [36] also temporally localized their video descriptions. Orthogonally, in an effort to increase the complexity of descriptions, TACoS multi-level [37] expanded the TACoS [36] dataset to include paragraph descriptions to instructional cooking videos. However, their dataset is constrained in the “cooking” domain and contains in the order of a 100 videos, making it un-

suitable for dense-captioning of events as the models easily overfit to the training data.

Our dataset, ActivityNet Captions, aims to bridge these three orthogonal approaches by temporally annotating long videos while also building upon the complexity of descriptions. ActivityNet Captions contains videos that an average of 180s long with the longest video running to over 10 minutes. It contains a total of 100k sentences, where each sentence is temporally localized. Unlike TACoS multi-level, we have two orders of magnitude more videos and provide annotations for an open domain. Finally, we are also the first dataset to enable the study of concurrent events, by allowing our events to overlap.

### 7.2. Detailed dataset statistics

As noted in the main paper, the number of sentences accompanying each video is normally distributed, as seen in Figure 6. On average, each video contains  $3.65 \pm 1.79$  sentences. Similarly, the number of words in each sentence is normally distributed, as seen in Figure 7. On average, each sentence contains  $13.48 \pm 6.33$  words, and each video contains  $40 \pm 26$  words.

There exists interaction between the video content and the corresponding temporal annotations. In Figure 8, the number of sentences accompanying a video is shown to be positively correlated with the video’s length: each additional minute adds approximately 1 additional sentence description. Furthermore, as seen in Figure 9, the sentence descriptions focus on the middle parts of the video more than the beginning or end.

When studying the distribution of words in Figures 10 and 11, we found that ActivityNet Captions generally focuses on people and the actions these people take. However,

Dataset	Domain	# videos	Avg. length	# sentences	Des.	Loc. Des.	paragraphs	overlapping
UCF101 [45]	sports	13k	7s	-	-	-	-	-
Sports 1M [21]	sports	1.1M	300s	-	-	-	-	-
Thumos 15 [15]	sports	21k	4s	-	-	-	-	-
HMDB 51 [25]	movie	7k	3s	-	-	-	-	-
Hollywood 2 [28]	movie	4k	20s	-	-	-	-	-
MPII cooking [40]	cooking	44	<b>600s</b>	-	-	-	-	-
ActivityNet [4]	human	20k	<b>180s</b>	-	-	-	-	-
MPII MD [39]	movie	68k	4s	68,375	✓	-	-	-
M-VAD [47]	movie	49k	6s	55,904	✓	-	-	-
MSR-VTT [55]	<b>open</b>	<b>10k</b>	20s	<b>200,000</b>	✓	-	-	-
MSVD [6]	<b>human</b>	<b>2k</b>	10s	<b>70,028</b>	✓	-	-	-
YouCook [7]	cooking	88	-	2,688	✓	-	-	-
Charades [43]	<b>human</b>	<b>10k</b>	30s	16,129	✓	-	-	-
KITTI [12]	driving	21	30s	520	✓	✓	-	-
TACoS [36]	cooking	127	<b>360s</b>	11,796	✓	✓	-	-
TACoS multi-level [37]	cooking	127	<b>360s</b>	52,593	✓	✓	✓	-
ActivityNet Captions (ours)	<b>open</b>	<b>20k</b>	<b>180s</b>	<b>100k</b>	✓	✓	✓	✓

Table 4: Compared to other video datasets, ActivityNet Captions contains long videos with a large number of sentences that are all temporally localized and is the only dataset that contains overlapping events. (Loc. Des. shows which datasets contain temporally localized language descriptions. Bold fonts are used to highlight the nearest comparison of our model with existing models.)

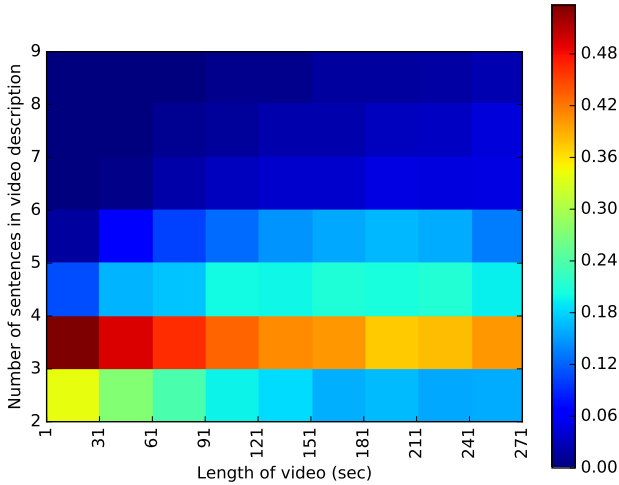


Figure 8: Distribution of number of sentences with respect to video length. In general the longer the video the more sentences there are, so far on average each additional minute adds one more sentence to the paragraph.

we wanted to know whether ActivityNet Captions captured the general semantics of the video. To do so, we compare our sentence descriptions against the shorter labels of ActivityNet, since ActivityNet Captions annotates ActivityNet videos. Figure 16 illustrates that the majority of videos in ActivityNet Captions often contain ActivityNet’s labels in

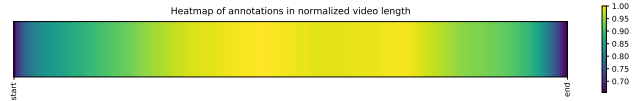


Figure 9: Distribution of annotations in time in ActivityNet Captions videos, most of the annotated time intervals are closer to the middle of the videos than to the start and end.

at least one of their sentence descriptions. We find that the many entry-level categories such as *brushing hair* or *playing violin* are extremely well represented by our captions. However, as the categories become more nuanced, such as *powerbocking* or *cumbia*, they are not as commonly found in our descriptions.

### 7.3. Dataset collection process

We used Amazon Mechanical Turk to annotate all our videos. Each annotation task was divided into two steps: (1) Writing a paragraph describing all major events happening in the videos in a paragraph, with each sentence of the paragraph describing one event (Figure 12; and (2) Labeling the start and end time in the video in which each sentence in the paragraph event occurred (Figure 13). We find complementary evidence that workers are more consistent with their video segments and paragraph descriptions if they are asked to annotate visual media (in this case, videos) using natural language first [23]. Therefore, instead of asking workers to

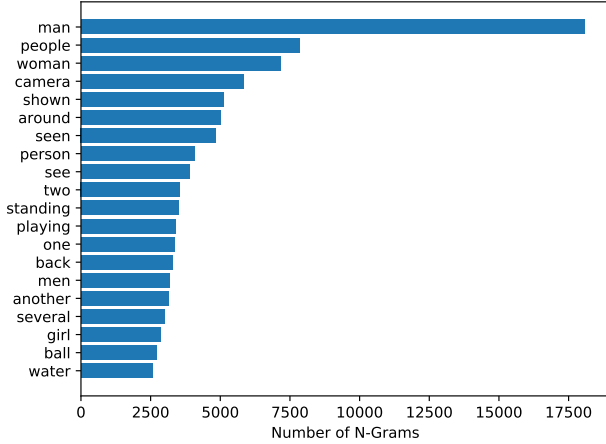


Figure 10: The most frequently used words in ActivityNet Captions with stop words removed.

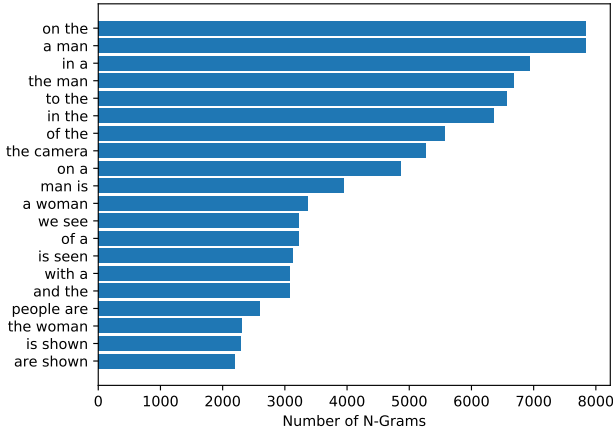


Figure 11: The most frequently used bigrams in ActivityNet Captions .

segment the video first and then write individual sentences, we asked them to write paragraph descriptions first.

Workers are instructed to ensure that their paragraphs are at least 3 sentences long where each sentence describes events in the video but also makes a grammatically and semantically coherent paragraph. They were allowed to use co-referencing words (ex, *he*, *she*, etc.) to refer to subjects introduced in previous sentences. We also asked workers to write sentences that were at least 5 words long. We found that our workers were diligent and wrote an average of 13.48 number of words per sentence. Each of the task and examples (Figure 14) of good and bad annotations.

Workers were presented with examples of good and bad annotations with explanations for what constituted a good paragraph, ensuring that workers saw concrete evidence of what kind of work was expected of them (Figure 14). We

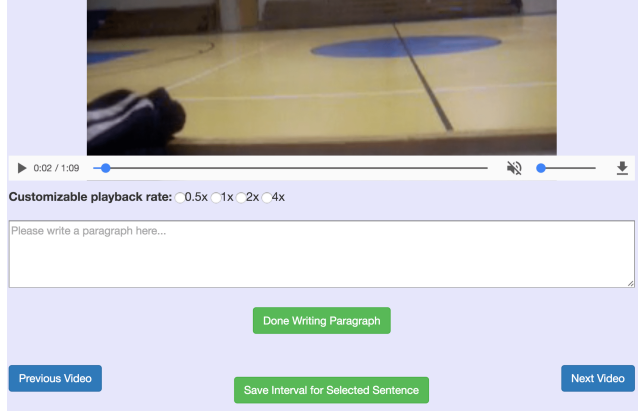


Figure 12: Interface when a worker is writing a paragraph. Workers are asked to write a paragraph in the text box and press "Done Writing Paragraph" before they can proceed with grounding each of the sentences.

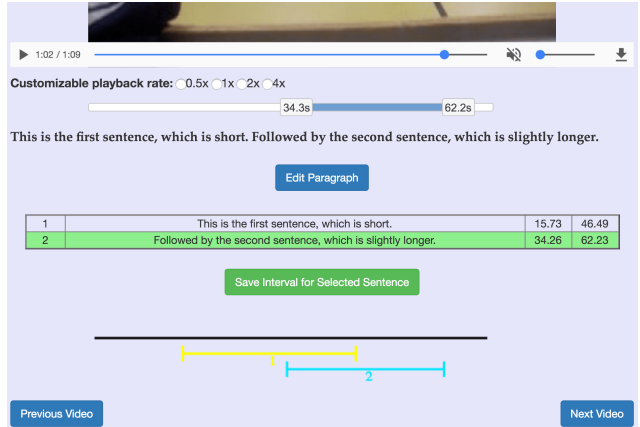


Figure 13: Interface when labeling sentences with start and end timestamps. Workers select each sentence, adjust the range slider indicating which segment of the video that particular sentence is referring to. They then click save and proceed to the next sentence.

paid workers \$3 for every 5 videos that were annotated. This amounted to an average pay rate of \$8 per hour, which is in tune with fair crowd worker wage rate [41].

#### 7.4. Annotation details

Following research from previous work that show that crowd workers are able to perform at the same quality of work when allowed to video media at a faster rate [24], we show all videos to workers at 2X the speed, i.e. the videos are shown at twice the frame rate. Workers do, however, have the option to watching the videos at the original video speed and even speed it up to 3X or 4X the speed. We found, however, that the average viewing rate chosen by workers

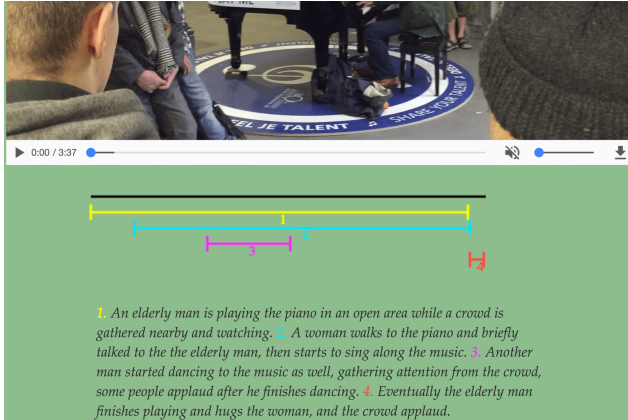


Figure 14: We show examples of good and bad annotations to workers. Each task contains one good and one bad example video with annotations. We also explain why the examples are considered to be good or bad.

was 1.91X while the median rate was 1X, indicating that a majority of workers preferred watching the video at its original speed. We also find that workers tend to take an average of 2.88 and a median of 1.46 times the length of the video in seconds to annotate.

At any given time, workers have the ability to edit their paragraph, go back to previous videos to make changes to their annotations. They are only allowed to proceed to the next video if this current video has been completely annotated with a paragraph with all its sentences timestamped. Changes made to the paragraphs and timestamps are saved when "previous video" or "next video" are pressed, and reflected on the page. Only when all videos are annotated can the worker submit the task. In total, we had 112 workers who annotated all our videos.

**Acknowledgements.** This research was sponsored in part by grants from the Office of Naval Research (N00014-15-1-2813) and Panasonic, Inc. We thank JunYoung Gwak, Timnit Gebru, Alvaro Soto, and Alexandre Alahi for their helpful comments and discussion.

## References

- [1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. **Social lstm: Human trajectory prediction in crowded spaces**. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–971, 2016.
- [2] C. Baldassano, J. Chen, A. Zadbood, J. W. Pillow, U. Hasson, and K. A. Norman. Discovering event structure in continuous narrative perception and memory. *bioRxiv*, page 081018, 2016.
- [3] O. Boiman and M. Irani. Detecting irregularities in images and in video. *International journal of computer vision*, 74(1):17–31, 2007.
- [4] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015.
- [5] F. Caba Heilbron, J. C. Niebles, and B. Ghanem. Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1914–1923, 2016.
- [6] D. L. Chen and W. B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL-2011)*, Portland, OR, June 2011.
- [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 *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
- [8] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.
- [9] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1491–1498. IEEE, 2009.
- [10] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem. Daps: **Deep action proposals for action understanding**. In *European Conference on Computer Vision*, pages 768–784. Springer, 2016.
- [11] A. Gaidon, Z. Harchaoui, and C. Schmid. Temporal localization of actions with actoms. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2782–2795, 2013.
- [12] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.
- [13] G. Gkioxari and J. Malik. Finding action tubes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 759–768, 2015.
- [14] D. B. Goldman, B. Curless, D. Salesin, and S. M. Seitz. Schematic storyboarding for video visualization and editing. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 862–871. ACM, 2006.
- [15] A. Ghorban, H. Idrees, Y.-G. Jiang, A. Roshan Zamir, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. <http://www.thumos.info/>, 2015.
- [16] M. Gygli, H. Grabner, and L. Van Gool. Video summarization by learning submodular mixtures of objectives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3090–3098, 2015.
- [17] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013.



- [18] J. Johnson, A. Karpathy, and L. Fei-Fei. Denscap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4565–4574, 2016.
- [19] S. Karaman, L. Seidenari, and A. Del Bimbo. Fast saliency based pooling of fisher encoded dense trajectories. In *ECCV THUMOS Workshop*, volume 1, 2014.
- [20] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.
- [21] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.
- [22] J. Krause, J. Johnson, R. Krishna, and L. Fei-Fei. A hierarchical approach for generating descriptive image paragraphs. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [23] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. In *International Journal on Computer Vision (IJCV)*, 2017.
- [24] R. A. Krishna, K. Hata, S. Chen, J. Kravitz, D. A. Shamma, L. Fei-Fei, and M. S. Bernstein. Embracing error to enable rapid crowdsourcing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 3167–3179. ACM, 2016.
- [25] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2556–2563. IEEE, 2011.
- [26] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.
- [27] W. Liu, T. Mei, Y. Zhang, C. Che, and J. Luo. Multi-task deep visual-semantic embedding for video thumbnail selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3707–3715, 2015.
- [28] M. Marszałek, I. Laptev, and C. Schmid. Actions in context. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2009.
- [29] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3, 2010.
- [30] B. Ni, V. R. Paramathayalan, and P. Moulin. Multiple granularity analysis for fine-grained action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 756–763, 2014.
- [31] J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *European conference on computer vision*, pages 392–405. Springer, 2010.
- [32] D. Oneata, J. Verbeek, and C. Schmid. Efficient action localization with approximately normalized fisher vectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2545–2552, 2014.
- [33] M. Otani, Y. Nakashima, E. Rahtu, J. Heikkilä, and N. Yokoya. Learning joint representations of videos and sentences with web image search. In *European Conference on Computer Vision*, pages 651–667. Springer, 2016.
- [34] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui. Jointly modeling embedding and translation to bridge video and language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4594–4602, 2016.
- [35] H. Pirsiavash and D. Ramanan. Parsing videos of actions with segmental grammars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 612–619, 2014.
- [36] M. Regneri, M. Rohrbach, D. Wetzels, S. Thater, B. Schiele, and M. Pinkal. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics (TACL)*, 1:25–36, 2013.
- [37] A. Rohrbach, M. Rohrbach, W. Qiu, A. Friedrich, M. Pinkal, and B. Schiele. Coherent multi-sentence video description with variable level of detail. In *German Conference on Pattern Recognition*, pages 184–195. Springer, 2014.
- [38] A. Rohrbach, M. Rohrbach, and B. Schiele. The long-short story of movie description. In *German Conference on Pattern Recognition*, pages 209–221. Springer, 2015.
- [39] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele. A dataset for movie description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [40] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1194–1201. IEEE, 2012.
- [41] N. Salehi, L. C. Irani, M. S. Bernstein, A. Alkhatib, E. Ogbe, K. Milland, et al. We are dynamo: Overcoming stalling and friction in collective action for crowd workers. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1621–1630. ACM, 2015.
- [42] C. Schuld, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36. IEEE, 2004.
- [43] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, 2016.
- [44] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5179–5187, 2015.
- [45] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.
- [46] Y. Tian, R. Sukthankar, and M. Shah. Spatiotemporal deformable part models for action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2642–2649, 2013.

- [47] 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.
- [48] A. Vahdat, B. Gao, M. Ranjbar, and G. Mori. A discriminative key pose sequence model for recognizing human interactions. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1729–1736. IEEE, 2011.
- [49] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4534–4542, 2015.
- [50] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko. Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*, 2014.
- [51] L. Wang, Y. Qiao, and X. Tang. Action recognition and detection by combining motion and appearance features. *THU-MOS14 Action Recognition Challenge*, 1:2, 2014.
- [52] L. Wang, Y. Qiao, and X. Tang. Video action detection with relational dynamic-poselets. In *European Conference on Computer Vision*, pages 565–580. Springer, 2014.
- [53] W. Wolf. Key frame selection by motion analysis. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, volume 2, pages 1228–1231. IEEE, 1996.
- [54] H. Xu, S. Venugopalan, V. Ramanishka, M. Rohrbach, and K. Saenko. A multi-scale multiple instance video description network. *arXiv preprint arXiv:1505.05914*, 2015.
- [55] J. Xu, T. Mei, T. Yao, and Y. Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5288–5296, 2016.
- [56] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, volume 14, pages 77–81, 2015.
- [57] 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*, volume 5, page 6, 2015.
- [58] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, pages 379–385. IEEE, 1992.
- [59] H. Yang, B. Wang, S. Lin, D. Wipf, M. Guo, and B. Guo. Unsupervised extraction of video highlights via robust recurrent auto-encoders. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4633–4641, 2015.
- [60] L. Yang, K. Tang, J. Yang, and L.-J. Li. Dense captioning with joint inference and visual context. *arXiv preprint arXiv:1611.06949*, 2016.
- [61] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE international conference on computer vision*, pages 4507–4515, 2015.
- [62] T. Yao, T. Mei, and Y. Rui. Highlight detection with pairwise deep ranking for first-person video summarization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 982–990, 2016.
- [63] S. Yeung, A. Fathi, and L. Fei-Fei. Videoset: Video summary evaluation through text. *arXiv preprint arXiv:1406.5824*, 2014.
- [64] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4584–4593, 2016.
- [65] H. J. Zhang, J. Wu, D. Zhong, and S. W. Smoliar. An integrated system for content-based video retrieval and browsing. *Pattern recognition*, 30(4):643–658, 1997.

	<b>Ground Truth</b> A man sits with his dog in the grass and holds out discs in his hands. The man balances his dog on his feet then throws Frisbee discs for him. The man spins his dog and holds it in his arms. Different trainers throw Frisbee discs for the dogs to catch while performing tricks. A woman throws discs to her dog that jumps from her back. The woman throws multiple discs in a row for her dog to catch.	<b>No Context</b> A man is seen speaking to the camera while holding a dog around him. The woman continues to swing around with the frisbee as well as performing tricks. The man then begins to do tricks with the dog while the camera follows him. A woman is seen walking out onto a field with a dog. The dog jumps off the girl and the dog jumps to the dog. The dog jumps off the girl and the dog jumps to the dog.	<b>With Context</b> A man is seen speaking to the camera while standing in a field with a dog. The dog is seen in several clips performing tricks with his dog and running all around the yard. The man then begins walking around with a frisbee. The dog runs around in circles on the field with the dog. The dog runs around in circles on the field with the frisbee. The dog runs around in circles on the grass as he chases the frisbee.
	<b>Ground Truth</b> A man is standing outside holding a black tile. He starts putting the tile down on the ground. He cuts the tile with a red saw. He sets chairs and flowers on the tile.	<b>No Context</b> a man is seen speaking to the camera while holding up a tool and begins to cut. the man then puts a on the floor and begins putting into the tire and. the man then puts a on the floor and begins putting tiles on the sides and. a person is seen pushing a puck down a floor with a rag and showing the camera.	<b>With Context</b> a man is seen speaking to the camera while holding up a bucket and begins painting the wall. a man is seen kneeling down on a roof and begins using a tool on the carpet. a man is seen speaking to the camera and leads into him holding knives and sharpening a board . the person then walks around the table and begins painting the fence.
	<b>Ground Truth</b> A little girl performs gymnastics jumping and flipping in the air. The little girl performs three back flips in the air, after she jumps. The girl flips but she falls, then she stands and does cartwheels and continues doing flips and dancing.	<b>No Context</b> A girl in a black shirt is standing on a mat. A girl in a black shirt is standing on a mat. A girl in a red shirt is standing in a large room in a large gymnasium.	<b>Full Context</b> The girl then begins flipping around the beam and ends by jumping off the side and walking away. The girl then flips herself over her feet and does several back flips on the mat. The girl then flips herself over her feet and does several flips and tricks.

Figure 15: More qualitative dense-captioning captions generated using our model. We show captions with the highest overlap with ground truth captions.

