Video Captioning of Future Frames

Mehrdad Hosseinzadeh[1] and Yang Wang[1,2]
University of Manitoba[1], Huawei Technologies Canada[2]
{mehrdad, ywang}@cs.umanitoba.ca

Abstract

Being able to anticipate and describe what may happen in the future is a fundamental ability for humans. Given a short clip of a scene about "a person is sitting behind a piano", humans can describe what will happen afterward, i.e. "the person is playing the piano". In this paper, we consider the task of captioning future events to assess the performance of intelligent models on anticipation and video description generation tasks simultaneously. More specifically, given only the frames relating to an occurring event (activity), the goal is to generate a sentence describing the most likely next event in the video. We tackle the problem by first predicting the next event in the semantic space of convolutional features, then fusing contextual information into those features, and feeding them to a captioning module. Departing from using recurrent units allows us to train the network in parallel. We compare the proposed method with a baseline and an oracle method on the ActivityNet-Captions dataset. Experimental results demonstrate that the proposed method outperforms the baseline and is comparable to the oracle method. We perform additional ablation study to further analyze our approach.

1. Introduction

In this paper, we consider the problem of generating captions for future frames in a video. This problem is related to video captioning. But there are important differences as well. In standard video captioning, all frames in the entire video are observed. But in our problem setting, we consider an online setting where we only have access to the frames observed so far. Our goal is to generate captions for future frames that are not observed yet.

Humans have the amazing ability to anticipate the future based on current events (see Fig. 1). Given a short clip of an event happening now, humans can easily anticipate and describe what will most likely happen next. For example, after observing that *"a coach is advising a weight lifting athlete"*, it is easy for us to anticipate that later on *"the athlete will lift the weight again after the advice"* (Fig. 1). The goal of this paper is to enable intelligent agents to have similar capabilities. Generating captions of future frames also has many important real-world applications. For example, consider the application of assisting visually impaired people. If we can have a system that can automatically generate captions of future events based on the observed visual scenes, the person will be able to anticipate possibly dangerous events in the future and take action appropriately.

Generating captions for future frames is a challenging problem. Since the algorithm does not have access to future frames, it needs to understand the semantic information of the current scene and accurately predict the future. Inspired by recent work on future semantic segmentation [20], we develop our approach by predicting the feature maps of future frames based on observed frames. Then we can use a standard captioning module to generate the captions based on the predicted feature maps of the future frames. we demonstrate superior performance over the baselines on the challenging ActivityNet-Captions dataset [14].

The contributions of the paper are manifold. First, we take tackle the problem of video captioning for upcoming future frames in a video in two different settings: general, and conditional captioning. Compared with standard video captioning, this new problem setting is closer to many real-world applications such as assistive technologies for the visually impaired. Second, we propose a novel approach to this problem. Our approach is based on predicting the feature maps of future frames, rather than directly generating pixel values. Finally, our experimental results demonstrate the effectiveness of the proposed approach compared with other baselines.

2. Related Work

Future frame captioning is related to several lines of research in computer vision, including image/video captioning, future prediction, and moment/event detection in videos. In image captioning, the goal is to generate a sentence describing an input image. Existing image captioning models often use recurrent neural networks (RNNs), specifically the long-short term memory (LSTM) variant [10]. Karpathy et al. [12] tackle the problem using a combina-

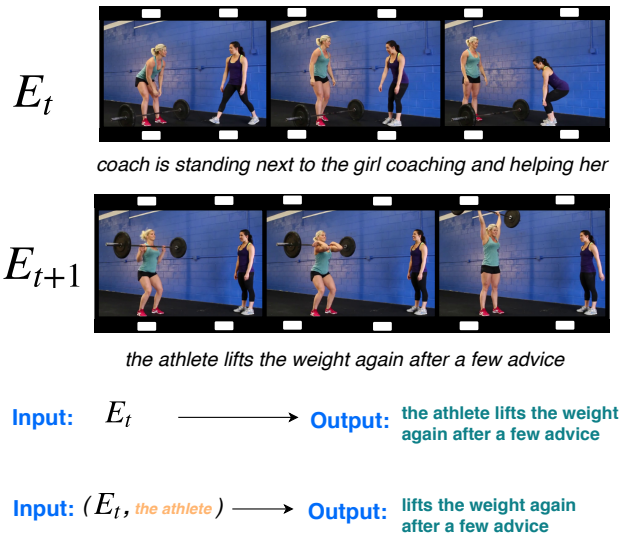

Figure 1. Given a sequence of frames of what is happening (E_t), the task is to anticipate what will happen in the next event (future) and describe it using a sentence. First row: the general version of the problem in which the task is to generate all the words in the caption for the next event. Second row: conditional future captioning where the task is to take the current event as well as a noun phrase defining the actor of the next event (*e.g.* the athlete in this case) and describe what the actor will do in the next event.

tion of CNNs for extracting features from the image and an RNN for generating the caption. Some works [32, 2, 18] use an attention mechanism to focus on visual features from different parts of the input image when generating each word.

There has also been lots of work on captioning in the video domain video [14, 29, 28, 33, 34]. The method in [28] utilizes LSTM units in an encoder-decoder fashion for the video captioning task. It extracts both appearance and optical flow features of frames, then feds them through their proposed model to generate captions. Inspired by the success of self-attention [26] and transformer networks, [38, 39] propose end-to-end video dense captioning systems by establishing a more explicit relationship between visual and textual modalities. Another attempt to densely describe a video is made by [7]. It uses a cycle-consistency scheme to train the network without the corresponding temporal annotations of events in the video. Zhang et al. [36] further utilize the syntax of the sentence to generate more plausible captions for videos. All of these approaches assume that we have access to the entire video. In contrast, we consider the case where the future frames are not available.

Our work is also related to *future prediction* tasks in computer vision. Felsen et al. [8] study the problem of predicting the players' next moves in water polo and basketball videos. However, their approach is constrained to the special case of those sports and a limited set of moves. Additionally, there have been proposed approaches that aim to predict the future in a pixel-level space. Using generative adversarial networks, [1] has established a framework to predict the next frame(s) directly in RGB space based on what has been the sequence of frames so far. Other works [19, 20] have considered predicting future semantic or instance segmentation of future frames.

Another line of related work is on detecting important moments/events in a video. Buch et al. develop the SST method [5] which uses sliding windows over the input video through a Gated Recurrent Unit (GRU). The computational complexity is reduced in their model by avoiding overlapping windows. In recent work, Li et al. [15] propose a method to detect the moments and caption them using a cross-module loss function. Xiong et al. [31] propose a progressive video description generator which sequentially takes frames for processing, and produces a caption as it moves forward along the temporal dimension. After processing all the frames, the produced captions are further refined by another module to add more coherency. Recent work in [24] uses a proposal-free approach that eliminates the need for costly region proposal operations.

3. Problem Statement

Anticipation is the ability to inference across space, time, causality, etc. [23]. It is considered to be a fundamental capability for an intelligent entity [9]. It allows humans to partially observe a scene and describe what may happen afterward. For instance, given a short clip of "*she opened the hood of the car*", we can describe the next possible scene (event) which could be "*she then examined the engine*" [35].

We define an event in a video as a number of consecutive frames in a video clip that capture an action being performed. This notion of an "*event*" is consistent with the definition in [14] as well. Our goal is to generate captions for future events that are not observed yet. We consider two problem settings for future frame captioning. In the first problem setting (which we call the *general case*), we are given a sequence of frames $\{f_i^t\}_{i=1}^v$ representing an event in the video at time-stamp t, where f_i^t denotes the i-th frame within the t-th event in a video. Note that the number of frames v of an event can be variable for different events. The goal is to generate a sentence $\{w_i^{t'}\}_{i=1}^m$ describing what is happening in the next important event in the video at time-stamp t' (where $t < t'$). Here m represents the number of words in the sentence and $w_i^{t'}$ is a word in the sentence. In this setup, the model should solely rely on what is happening at the current moment in the video. It needs to infer what is the *subject* of the upcoming event and describe what the subject would do next.

However, due to the uncertainty of the future, the cur-

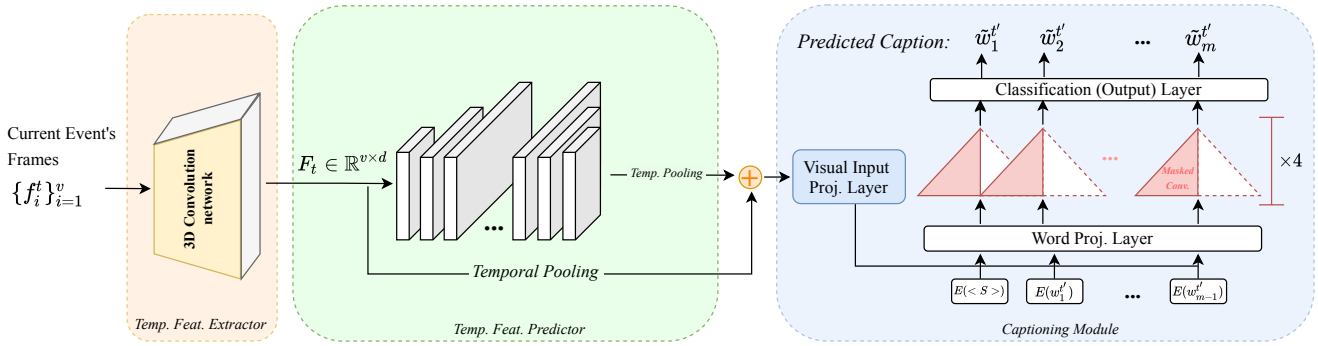

Figure 2. After training TFP, it is plugged into the main pipeline for the next stage of training. To train the main pipeline, a pre-trained C3D network computes the convolution features for each event (yellow box). Extracted features are then fed into the TFP to obtain the predicted features for the next event (green box). After temporal pooling, the predicted features are combined with the contextual features coming from the current event ($\oplus$) and are input to the captioning module using a projection layer. The captioning module adds the visual features to the word embedding features for each input word, applies n layers of mask convolution to effectively increase the receptive field for each word. The next word is generated by using softmax on top of the classification layer in the captioning module (blue box).

rent event could logically be followed by several different events with different subjects. For example, consider the case where the current event is "*the person blows the leaves from a grass area using the blower*". Potentially, both "***the blower** is seen up close*." and "***the person** then walks away from the camera.*" make sense even to humans to be the next possible event in the video.

To take into consideration this uncertainty, we take inspiration from some work in the NLP community [35] and propose the second problem setting. In this problem setting (which we call the *conditional case*), we have access to the visual information for the current event in the video as well as the actor (subject) for the next event. Our goal is to generate the caption for the next time-stamp, given the current visual information and the noun phrase representing the actor of interest for the next event.

4. Proposed Approach

One possible way of future frame prediction is to first predict future frames themselves, then apply a captioning model on the predicted frames. However, directly predicting the pixel values of future frames is challenging. Some recent work in semantic segmentation [19, 20] suggests that predicting convolutional features of future frames is a better choice than predicting raw pixel values. Following these previous works, we also focus on forecasting the convolutional features for the next event and generate captions based on the predicted features. Our proposed method consists of three major modules: a 3D convolution network as a backbone, a feature predictor, and a captioning module (Fig. 2).

Temporal Feature Extractor. 3D convolutional networks [14, 7] have been popular in video understanding tasks. In this work, we use the 3D convolutional model proposed by [11] as our feature extractor backbone. Given a sequence of raw frames for the current event $\{f_i^t\}_{i=1}^v$, this network processes the sequence and outputs a $F_t \in \mathbb{R}^{v \times d}$ feature map where v is the number of frames and d is the feature dimension. These features are used as input to the following modules.

Temporal Feature Predictor. Given the feature map F_t from the t-th event, the goal of this module is to predict the feature map $F_{t'}$ of the next event t' ($t < t'$). We first describe this module during the training stage. Each training instance consists of a pair of feature maps $(F_t, F_{t'})$ extracted by the temporal feature extractor for two adjacent events (t, t') from a training video. Note that the temporal dimensions of F_t and $F_{t'}$ can be different, i.e. $F_t \in \mathbb{R}^{v \times d}, F_{t'} \in \mathbb{R}^{v' \times d}$ where $v \neq v'$. The goal of the temporal feature predictor (TFP) module is to predict $F_{t'}$ given F_t.

This module uses a temporal convolution on F_t to produce a new feature map. During training, we learn the parameters of this temporal convolution layer, so that the predicted new feature map matches $F_{t'}$. In order to handle the case where F_t and $F_{t'}$ have different temporal dimensions (i.e. $v \neq v'$), we use a dynamic adaptive pooling (DAP) layer at the beginning of the TFP pipeline (see Fig. 3) which shrinks or enlarges the temporal dimension, depending on the length of the next event. In other words, this module can be written as:

$$G_{t'} = Conv(DAP(F_t)), \text{where } G \in \mathbb{R}^{v' \times d} \quad (1)$$

Here $DAP(\cdot)$ is the dynamic adaptive pooling operation that maps the temporal dimension of the input to t', while

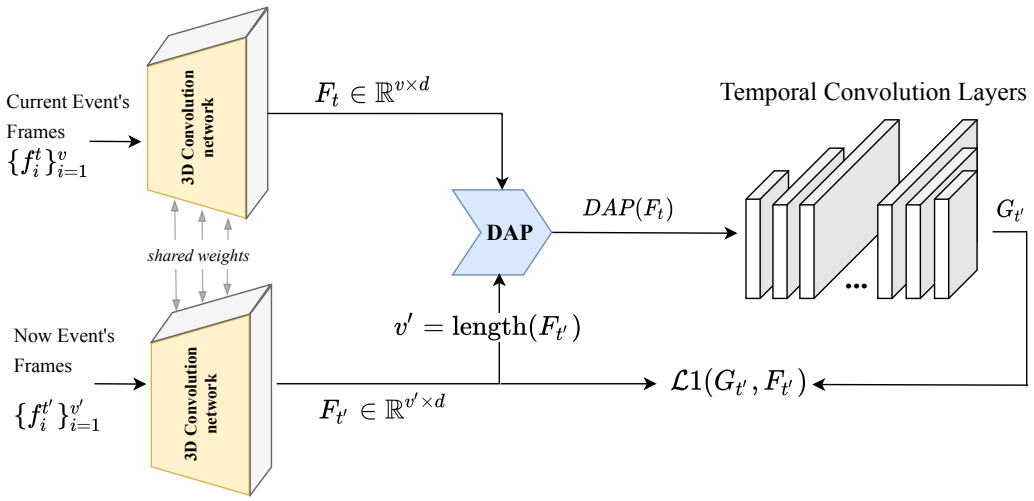

Figure 3. Training of TFP module on a pair of consecutive events at t and t' ($t < t'$). For each event, 3D convolution features are computed using a pre-trained C3D network F_t and $F_{t'}$. A dynamic adaptive pooling layer (DAP) takes the first event's features, F_t and the temporal length of the next event, v', and resizes the first event temporally to match the temporal dimension of the next event. The features are fed into a network consisting of several temporal convolution layers (grey blocks) to obtain the predicted features for the next event. The temporal feature predictor network maintains the temporal dimension throughout the layers. Grey dashed arrows indicate weight sharing.

$Conv(\cdot)$ is the temporal convolution operation. Note that DAP only operates during the training process. During testing, v' is unknown and thus we remove DAP from the pipeline. As a result, the generated features during testing have a temporal length of v. Note that DAP does not have any weights to learn, omitting it from the testing pipeline does not harm the performance of the trained module.

Captioning Module. While it is possible to directly feed the captioning module with the predicted feature vector $G_{t'}$ of the next event, this vector does not carry any information about the context in which the next event should be described by the captioning module. Having the context added to the visual information vector enables this module to describe the next event more accurately. The feature map of the current event can be considered a source of context, so we propose to combine it with the predicted feature map of the next event as follows:

$$G_{t'}^{final} = \lambda \cdot \mathcal{AVG}(F_t) \oplus (1 - \lambda) \cdot \mathcal{AVG}(G_{t'}) \quad (2)$$

where $\oplus$ is the element-wise summation, $\mathcal{AVG}(\cdot)$ is the average pooling along the temporal dimension, and $0 \leq \lambda \leq 1$ is a hyper-parameter of the model which controlls the amount of context to be fused into the next event's features. $G_{t'}^{final}$ is considered as the visual information going through the captioning module.

Although LSTM-based models have been widely used in tasks relating to joint vision and natural language processing [33, 22, 37, 4, 30], they fall short in two aspects. First, it is a common issue among the LSTM-based models that as the length of the sentence becomes longer, the performance of these models drops significantly [26, 2]. Second, LSTMs (and other recurrent units) are not easily parallelizable since the input for each time step needs to be calculated before the unit can move on to the next one. Lately, a number of approaches have been proposed to address these shortcomings by either using only self-attention layers [26, 38] or convolution layers [2] for sequence to sequence tasks involving natural language processing. Using either of these approaches can make the training easily parallelizable.

For the captioning module, we take advantage of the convolutional captioning module proposed by [2] because of its great performance on the image captioning task. Note that the proposed method is not limited by any specific captioning architecture. This module used in this work can be easily substituted with any existing alternative. We adopt a similar architecture that has been used in [2] with some modifications. In [2], the visual input to the captioning module is a vector of 4096 features coming from the FC7 layer of a VGG-16 network [25]. But in our case, we take the feature G from the temporal feature predictor and perform an average pooling over the dimension to obtain an d-dimensional feature vector. This feature vector is then used as the input to the captioning module.

Training and Inference. To train our model, we first extract convolution features for each event in a training video offline using a pre-trained 3D model. This results in a

$v \times d$ feature representation for each event where we use $v = T/16$, $d = 500$ in the experiments. Here T is the number of frames in the video clip corresponding to the event.

Then the feature extractor model is trained using the offline-computed features from the 3D network. As mentioned earlier, this network receives a $v \times d$ representation of the current event. It temporally resizes it to $v' \times d$ on the fly and generates the predicted features of the next event, $G_{t'} \in v' \times d$. To train this network we use a $\mathcal{L}1$ loss function between the predicted features of the next event and the pre-extracted ones:

$$\mathcal{L}_{TFP} = \sum_{i=1}^{v'} \sum_{j=1}^{d} |G_{t'}(i,j) - F_{t'}(i,j)| \tag{3}$$

Once this module is trained, it is plugged into the proposed network and the entire network is trained solely using a standard cross-entropy captioning loss. Note that when adding the temporal feature predictor module into the pipeline, the dynamic adaptive pooling layer (DAP) is detached and is no longer used.

The captioning module is first trained on the MSCOCO [17] dataset which has a total number of 9.2K words in its vocabulary. The weights of the pre-trained model are then used to initialize the weights of the captioning module in our case. Since the vocabulary size is 6K in our problem, we have to replace the first and last layers of the captioning module, *i.e.* the word embedding and word outputting layers in Fig. 2. Therefore for the new outputting layer (of size 6K), weights are initialized with a normal distribution where the mean and standard deviation are calculated based on the learned weights on MSCOCO. For the embedding layer, on the other hand, the weights are initialized randomly sampled from a normal distribution with *mean=0* and *std=0.1*.

Putting everything together, now the entire network is trained using a cross-entropy loss defined on the probability of predicted words in the sentence. At each position i in the sentence and for each word w, the probability is defined as $p(\tilde{w}_i | w < i, I)$ where $y < i$ are the ground truth (GT) words in positions before i and I is the projected visual features.

Since during training we have access to the GT word at each position, it is feasible to train the network in parallel. Nonetheless, inference happens sequentially since the prediction of each word depends on the previous predicted words, $p(\tilde{w}_i | \tilde{w} < i, I)$. Sentence generation starts with feeding a special token $< S >$ indicating the beginning of the sentence and is continued until the end-of-sentence token $< EOS >$ is generated (or reaching the maximum number of generated words).

4.1. Implementation Details

The feature extractor module consists of 7 temporal dilated convolution layers (except the first one), where each convolution layer is followed by a ReLU layer. The first 3 layers convert the feature depth to 1024 while preserving the temporal length. The next 4 layers return the feature depth to 500 again while retaining the temporal dimension of input (Fig. 3). Feature extractor is trained with an SGD optimizer with an initial learning rate of 0.001. The captioning module uses 3 layers of masked convolution with a kernel size of 5. Word projection and image projection layers map their input into a 300-d and 512-d space, respectively. RSMProp optimizer has been used with an initial learning rate of 5×10^{-5}, while the feature predictor module learns with $LR = 10^{-6}$ after plugging into the main pipeline. We experiment with the proposed approach with different λ and the results are reported in Table 3.

5. Experiments and Results

In this section, we first introduce the datasets in Sec. 5.1 and the evaluation metrics in Sec. 5.2. We then introduce methods used for comparison in Sec. 5.3. We present the experimental results in Sec. 5.4. Finally, We then more precisely examine our approach through ablation study. Finally, we perform ablation studies in Sec. 5.5.

5.1. Datasets

We use the ActivityNet-Captions dataset [14] and the SWAG-AF dataset [35] in the experiments. The ActivityNet dataset [6] is a large-scale benchmark for video understanding. Krishna et al. [14] have expanded the dataset by providing temporal annotations and descriptions for the events in the video to form the ActivityNet-Captions dataset. There are about 20K Youtube videos in the dataset with an average of 3.65 events per video. Videos cover a broad spectrum of human activities. There are about 100,000 events. Each event is temporally annotated and described using a sentence. We use the standard train/val split in [14] for the ActivityNet-Captions dataset.

The SWAG-AF dataset [35] has been recently introduced for the NLI and entailment task. To test the performance on future conditional captioning tasks, we use SWAG-AF to extract the noun phrase representing the actor in the next event. The noun phrase is used as the initial words for the caption of the next event. Note that SWAG-AF contains the samples of the ActivityNet-Caption dataset. Thus, for the conditional case, we report the results on the intersection of ActivityNet-Captions and SWAG-AF. Moreover, since some of the events in the ActivityNet-Captions are highly overlapped with each other, we use the SWAG-AF dataset to filter out those events –as such events are not included in SWAG-AF.

Method	BLEU@2	BLEU@3	BLEU@4	CIDEr	METEOR	ROUGE-L
Baseline	7.87	3.08	1.32	12.77	7.24	17.76
Proposed method ($\lambda = 0$)	8.25	3.33	1.52	13.49	7.80	18.46
Proposed method ($\lambda = 0.5$)	**8.55**	**3.56**	**1.60**	**15.28**	**7.82**	**18.62**
oracle	*8.70*	*3.80*	*1.90*	*17.05*	*8.05*	*18.87*

Table 1. Performance of the proposed method compared to the baseline and oracle method on the first problem setting (i.e. general case). The hyperparameter λ controls the amount of visual contextual information added to the predicted features for the next event. Adding context substantially boosts the performance of the proposed method, especially CIDEr metric.

Method	BLEU@2	BLEU@3	BLEU@4	CIDEr	METEOR	ROUGE-L
Baseline	9.04	4.23	1.95	20.81	**7.17**	**18.57**
proposed method ($\lambda = 0.5$)	**9.09**	**5.01**	**2.76**	**26.55**	7.02	18.16
oracle	*9.38*	*4.6*	*2.30*	*27.17*	*7.45*	*19.08*

Table 2. Performance of the proposed method on the conditional future captioning task compared to the baseline and oracle methods. The hyperparameter λ controls the amount of visual contextual information added to the predicted features for the next event.

5.2. Evaluation Metrics

We evaluate the performance of our approach using several evaluation metrics, including BLEU@N, METEOR, ROUGE-L, and CIDEr. BLEU@N [21] is a family of methods that computes the precision of the generated caption using N-gram matching. METEOR [3] has more focus on the recall accuracy of the generated caption. ROUGE-L [16] measures the quality of the generated caption using the longest common subset between the predicted and the ground-truth sentence. CIDEr [27] is a newly-introduced metric and is reported to be very consistent with human judgments.

5.3. Compared Methods

We set up a baseline and an oracle method to compare the performance of the proposed method. For the baseline, we extracted the visual features for the current event using the feature extracted network (3D network). The features are then averaged along the temporal dimension and are fed into the captioning module. In other words, the caption decoder in this baseline has to caption the next event solely based on the features of the current event. To train the baseline, each event is accompanied by the consequent event's caption as the ground-truth.

We also set up an oracle method that has access to the next event and generates captions for future events based on the corresponding visual features. To do so, the oracle receives the extracted features for the next events, averages them across the frames to obtain a $500 - d$ vector which is then fed into the captioning module as the visual input. This oracle represents an upper bound for this task. We want the performance of our method to be as close as possible to this oracle. For the sake of fair comparison, we use the same backbone feature extractor and captioning modules in all methods (baseline, oracle, and ours). However, the proposed method is by no means constrained to these backbones. It can be used in conjunction with any video feature extractor and/or captioning module available in the literature.

5.4. Results

We first analyze the performance of our method in the first problem setting where the model is asked to generate the caption without having access to information about the next event's actor. Table 1 reports the performance on this task. The second row ($\lambda = 0$) is the case where the method does not take advantage of the context of the video. As seen in Table 1, the proposed method outperforms the baseline. But there is still a noticeable gap with the oracle's performance. When adding the visual context information to the predicted feature vector of the next event (third row in Table 1), the performance of the proposed method increases substantially.

Furthermore, we examine the performance of the proposed method against the baseline and oracle in the conditional future captioning task. In this task, during inference for each event e, the models have access to a ground-truth noun phrase for the next event $N_e = \{w_i\}_{i=1}^{n_e}$ which indicates the actor in the next event. Although at first glance, this task seems to be easier than the general case, Table 2 shows that it is still a challenging task. For this task, the proposed method outperforms the baseline by a large margin in terms of BLEU@3, BLEU@4, and more significantly in terms of CIDEr. Using more information about the future event, i.e. the next event's actor entity tends to boost almost every metric in Table 2 (compared with Table 1) except METEOR. For METEOR, it slightly decreases. We believe this

Method	BLEU@2	BLEU@3	BLEU@4	CIDEr	METEOR	ROUGE-L
Proposed method ($\lambda = 0$)	8.25	3.33	1.52	13.49	7.80	18.46
Proposed method ($\lambda = 0.35$)	8.47	3.51	1.57	14.80	7.93	18.62
Proposed method ($\lambda = 0.50$)	**8.55**	**3.56**	**1.60**	**15.28**	7.82	18.62
Proposed method ($\lambda = 0.65$)	8.51	3.50	1.53	15.05	**7.96**	**18.83**

Table 3. Performance of the proposed method in the first problem setting (i.e. general case) using different values of λ. By increasing λ and therefore injecting more contextual information, we obtain better results. However when λ becomes bigger than 0.50, the performance starts to drop.

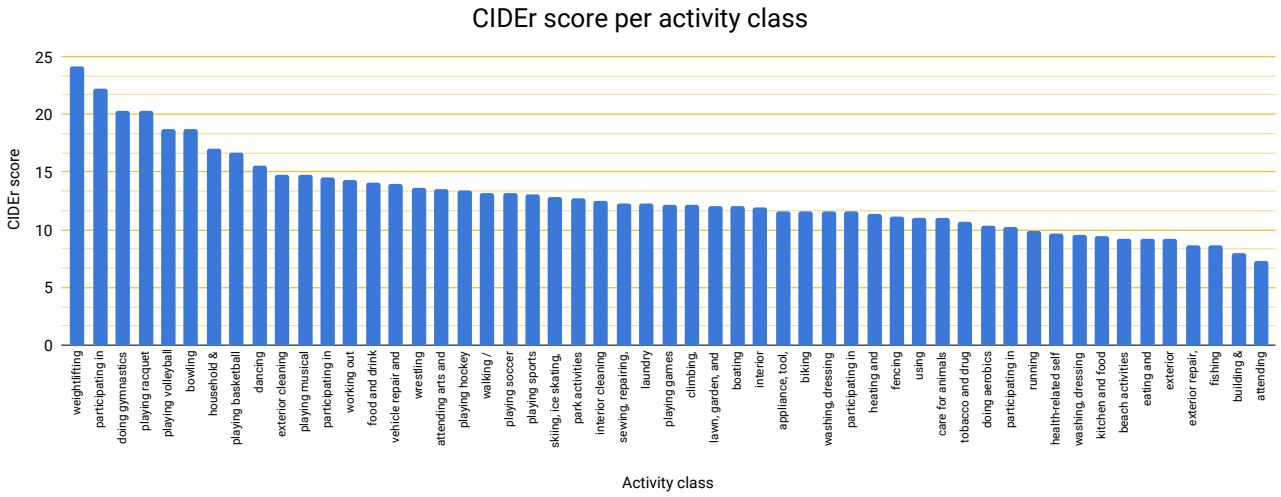

Figure 4. CIDEr score for different activity classes. While our proposed method works generally well, we found that it works best in the events relating to the sports and has moderate performance in more complex environments such as "*building & repairing furniture*", "*exterior repair, improvements, & decoration*", and "*exterior maintenance, repair, & decoration*".

is due to the nature of the METEOR score that sometimes fails to capture the similarity between two sentences as it is not originally introduced for captioning tasks [13]. The CIDEr score, on the other hand, is specifically designed to evaluate the captioning-related tasks and one can see a significant boost on the CIDEr score in Table 1 compared with Table 2. Fig. 5 provides some qualitative examples of our method on the ActivityNet-Caption validation set. The proposed method works well on the sport-related events and works moderately in complex scenes as it does in the third example.

5.5. Ablation Study

The hyperparameter λ in our method controls the relative contribution of the context information from the current event and the predicted features of the next event. We analyze the importance of λ and find the most suitable value for it through an ablation study. We split the original *training* set of ActivityNet-Captions dataset into two new disjoint sets of training and evaluation. The new training set has 80% of the original training samples and the new evaluation set inherits the remaining 20% samples. We then run our method on the new training set and measure the performance using the new evaluation set. Once we found the λ value that works best on the evaluation set (in this case $\lambda = 0.50$), the proposed method is trained again on the entire *original* training set and tested on the original validation set to obtain the results in Table 1 and Table 3.

Setting λ to 0.50 means that half of the visual information is coming from the current event while the other half is from the predicted features using the TFP module. In other words, we equally rely on our TFP module and the information that we have at hand about the current event. Table 3 clearly shows that this strategy yields the most appealing results. Interestingly adding more context information (e.g. $\lambda = 0.65$) is detrimental to the general performance of the method. Having $\lambda = 0.65$ causes the model to lose to the case where the λ is set to 0.50 in 5 out of 7 metrics, namely in BLEU@$\{2,3,4\}$, CIDEr, and METEOR metrics.

To further analyze the proposed method when captioning different activity types, we present the CIDEr score for each class of activity. In total there are 200 unique activity labels

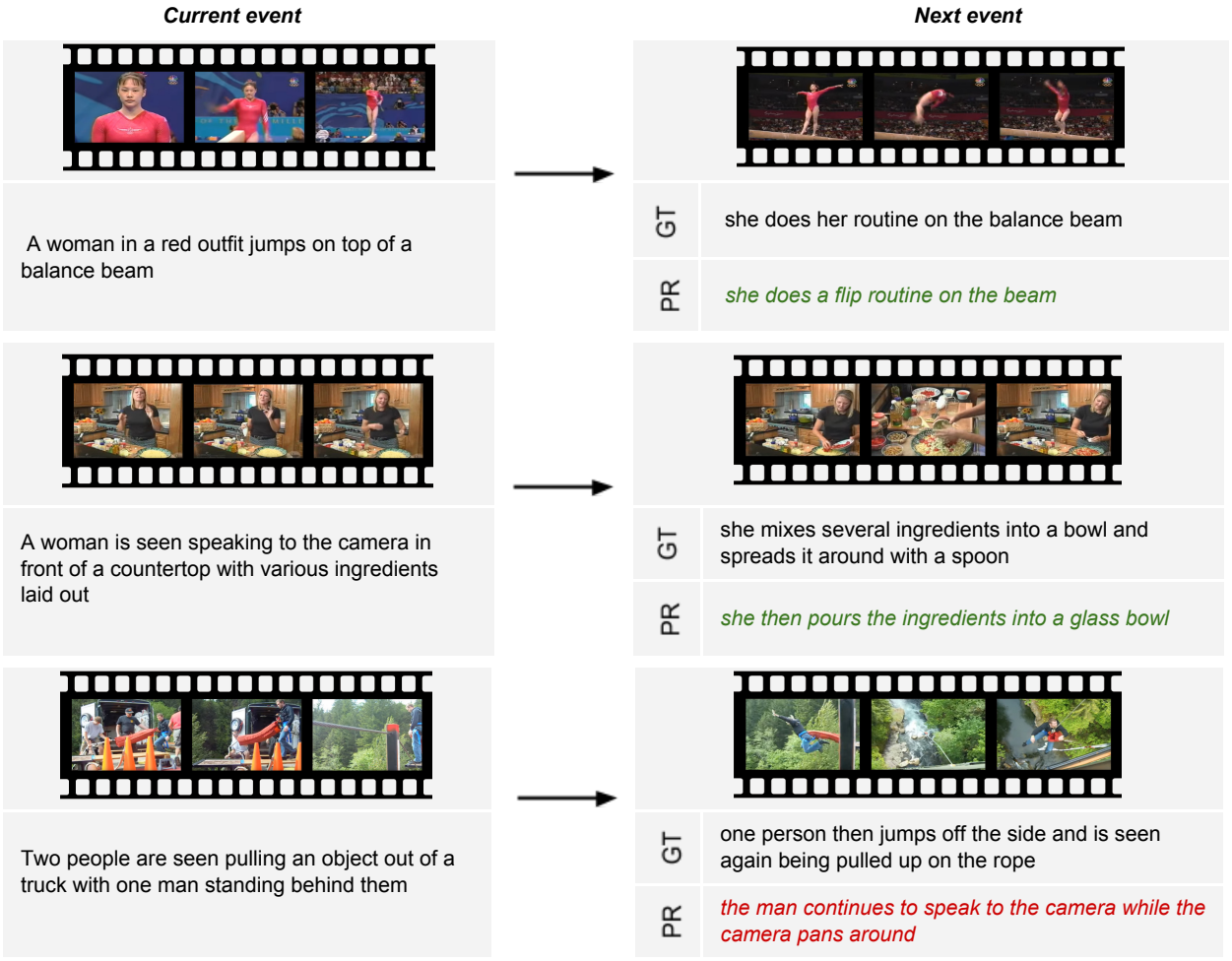

Figure 5. Qualitative examples. GT and PR are the ground-truth and predicted caption for the next event. In the first and second examples, the proposed method accurately captions the next event. But it fails to describe properly in the third example.

available on the ActivityNet V1.3 dataset. Based on their semantics, these activities are merged together to form 53 super activity groups. Following this taxonomy, we merge the event in the validation set according to their supergroup labels as well. We then use the best version of our method, *i.e.* $\lambda = 0.5$, for each of those groups. Finally, we compute the CIDEr score for each supergroup individually. Fig. 4 depicts the obtained result for this analysis.

We have found that our method is most effective for activities related to sports. The top 5 most accurately activity types are "*weightlifting*", "*participating in rodeo competitions*", "*doing gymnastics*", "*playing racquet*", and "*playing volleyball*". On the other hand, the least accurately activity types are "*building & repairing furniture*", "*fishing*", "*exterior repair, improvements, & decoration*", and "*exterior maintenance, repair, & decoration*". We believe this is due to the fact that these activities tend to have large variations in their environment and scene.

6. Conclusion

The ability to anticipate and describe what might happen next is a fundamental capability of human beings. In this paper, we have tackled the problem of captioning future frames in a video given the currently observed frames. We have proposed an architecture that first predicts the convolutional features for the next event, then fuses the features with the context features coming from the current event. Finally, it uses the fused feature to generate the caption. Our experimental results demonstrate that the proposed approach outperforms the baseline.

Acknowledgement: This work was funded by NSERC. We thank NVIDIA for donating some of the GPUs used in this work.

References

[1] Sandra Aigner and Marco Körner. Futuregan: Anticipating the future frames of video sequences using spatio-temporal 3d convolutions in progressively growing autoencoder gans. *arXiv preprint arXiv:1810.01325*, 2018.

[2] Jyoti Aneja, Aditya Deshpande, and Alexander G Schwing. Convolutional image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5561–5570. IEEE, 2018.

[3] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72. Association for Computational Linguistics, 2005.

[4] Marc Bolaños, Álvaro Peris, Francisco Casacuberta, Sergi Soler, and Petia Radeva. Egocentric video description based on temporally-linked sequences. *Journal of Visual Communication and Image Representation*, 50:205–216, 2018. Elsevier.

[5] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. Sst: Single-stream temporal action proposals. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6373–6382. IEEE, 2017.

[6] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–970. IEEE, 2015.

[7] Xuguang Duan, Wenbing Huang, Chuang Gan, Jingdong Wang, Wenwu Zhu, and Junzhou Huang. Weakly supervised dense event captioning in videos. In *Processings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 3063–3073. Curran Associates, Inc., 2018.

[8] Panna Felsen, Pulkit Agrawal, and Jitendra Malik. What will happen next? forecasting player moves in sports videos. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pages 3342–3351. IEEE, 2017.

[9] Jeff Hawkins and Sandra Blakeslee. *On intelligence: How a new understanding of the brain will lead to the creation of truly intelligent machines*. Macmillan, 2007.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. MIT Press.

[11] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2013. IEEE.

[12] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3128–3137. IEEE, 2015.

[13] Mert Kilickaya, Aykut Erdem, Nazli Ikizler-Cinbis, and Erkut Erdem. Re-evaluating automatic metrics for image captioning. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 199–209, 2017.

[14] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 706–715. IEEE, 2017.

[15] Yehao Li, Ting Yao, Yingwei Pan, Hongyang Chao, and Tao Mei. Jointly localizing and describing events for dense video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7492–7500. IEEE, 2018.

[16] Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 605–612. Association for Computational Linguistics, 2004.

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European conference on computer vision (ECCV)*, pages 740–755. Springer, 2014.

[18] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern recognition (CVPR)*, pages 375–383. IEEE, 2017.

[19] Pauline Luc, Camille Couprie, Yann Lecun, and Jakob Verbeek. Predicting future instance segmentation by forecasting convolutional features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 584–599. Springer, 2018.

[20] Pauline Luc, Natalia Neverova, Camille Couprie, Jakob Verbeek, and Yann LeCun. Predicting deeper into the future of semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 648–657. IEEE, 2017.

[21] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[22] Álvaro Peris, Marc Bolaños, Petia Radeva, and Francisco Casacuberta. Video description using bidirectional recurrent neural networks. In *Proceedings of the International Conference on Artificial Neural Networks*, pages 3–11. Springer, 2016.

[23] Alexander Riegler. The role of anticipation in cognition. In *Proceedings of the AIP Conference*, volume 573, pages 534–541. American Institute of Physics, 2001.

[24] Cristian Rodriguez, Edison Marrese-Taylor, Fatemeh Sadat Saleh, Hongdong Li, and Stephen Gould. Proposal-free temporal moment localization of a natural-language query in video using guided attention. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 2464–2473, 2020.

[25] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *In-*

ternational Conference on Learning Representations (ICLR), 2015.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008. Curran Associates, Inc., 2017.

[27] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575. IEEE, 2015.

[28] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4534–4542. IEEE, 2015.

[29] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1494–1504. Association for Computational Linguistics, 2015.

[30] Spencer Whitehead, Heng Ji, Mohit Bansal, Shih-Fu Chang, and Clare Voss. Incorporating background knowledge into video description generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3992–4001. Association for Computational Linguistics, 2018.

[31] Yilei Xiong, Bo Dai, and Dahua Lin. Move forward and tell: A progressive generator of video descriptions. *arXiv preprint arXiv:1807.10018*, 2018.

[32] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the International Conference on Machine learning (ICML)*, pages 2048–2057. The Journal of Machine Learning Research, 2015.

[33] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4507–4515. IEEE, 2015.

[34] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4584–4593. IEEE, 2016.

[35] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. Swag: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 93–104. Association for Computational Linguistics, 2018.

[36] Qi Zheng, Chaoyue Wang, and Dacheng Tao. Syntax-aware action targeting for video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[37] Luowei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J Corso, and Marcus Rohrbach. Grounded video description. *arXiv preprint arXiv:1812.06587*, 2018.

[38] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8739–8748. IEEE, 2018.

[39] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.