This CVPR 2020 workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Multi-modal Dense Video Captioning

Vladimir Iashin
Tampere University
vladimir.iashin@tuni.fi

Esa Rahtu
Tampere University
esa.rahtu@tuni.fi

Abstract

Dense video captioning is a task of localizing interesting events from an untrimmed video and producing textual description (captions) for each localized event. Most of the previous works in dense video captioning are solely based on visual information and completely ignore the audio track. However, audio, and speech, in particular, are vital cues for a human observer in understanding an environment. In this paper, we present a new dense video captioning approach that is able to utilize any number of modalities for event description. Specifically, we show how audio and speech modalities may improve a dense video captioning model. We apply automatic speech recognition (ASR) system to obtain a temporally aligned textual description of the speech (similar to subtitles) and treat it as a separate input alongside video frames and the corresponding audio track. We formulate the captioning task as a machine translation problem and utilize recently proposed Transformer architecture to convert multi-modal input data into textual descriptions. We demonstrate the performance of our model on ActivityNet Captions dataset. The ablation studies indicate a considerable contribution from audio and speech components suggesting that these modalities contain substantial complementary information to video frames. Furthermore, we provide an in-depth analysis of the ActivityNet Caption results by leveraging the category tags obtained from original YouTube videos. Code is publicly available: github.com/v-iashin/MDVC.

1. Introduction

The substantial amount of freely available video material has brought up the need for automatic methods to summarize and compactly represent the essential content. One approach would be to produce a short video skim containing the most important video segments as proposed in the *video summarization* task [25]. Alternatively, the video content could be described using natural language sentences. Such an approach can lead to a very compact and intuitive representation and is typically referred to as video captioning

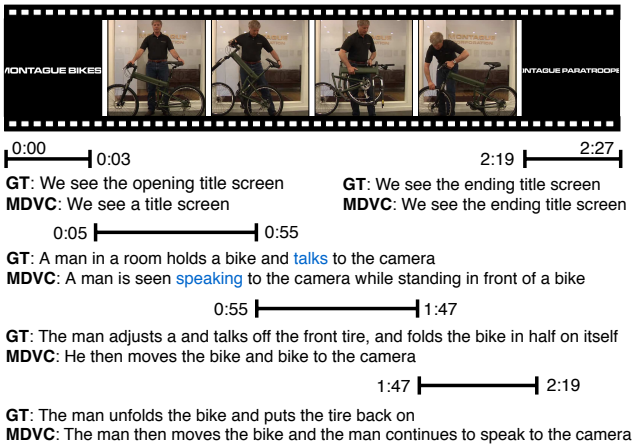

Figure 1. Example video with ground truth captions and predictions of Multi-modal Dense Video Captioning module (MDVC). It may account for any number of modalities, *i.e.* audio or speech.

in the literature [58]. However, producing a single description for an entire video might be impractical for long unconstrained footage. Instead, *dense video captioning* [24] aims, first, at temporally localizing events and, then, at producing natural language description for each of them. Fig. 1 illustrates dense video captions for an example video sequence.

Most recent works in dense video captioning formulate the captioning problem as a machine translation task, where the input is a set of features extracted from the video stream and the output is a natural language sentence. Thus, the captioning methods can be leveraged by recent developments in machine translation field, such as *Transformer* model [45]. The main idea in the transformer is to utilise *self-attention* mechanism to model long-term dependencies in a sequence. We follow the recent work [59] and adopt the transformer architecture in our dense video captioning model.

The vast majority of previous works are generating captions purely based on visual information [59, 48, 26, 28, 53, 30, 54]. However, almost all videos include an audio track, which could provide vital cues for video understanding. In particular, what is being said by people in the video, might make a crucial difference to the content description. For in-

stance, in a scene when someone knocks the door from an opposite side, we only see the door but the audio helps us to understand that somebody is behind it and wants to enter. Therefore, it is impossible for a model to make a useful caption for it. Also, other types of videos as instruction videos, sport videos, or video lectures could be challenging for a captioning model.

In contrast, we build our model to utilize video frames, raw audio signal, and the speech content in the caption generation process. To this end, we deploy *automatic speech recognition* (ASR) system [1] to extract time-aligned captions of *what is being said* (similar to subtitles) and employ it alongside with video and audio representations in the transformer model.

The proposed model is assessed using the challenging ActivityNet Captions [24] benchmark dataset, where we obtain competitive results to the current state-of-the-art. The subsequent ablation studies indicate a substantial contribution from audio and speech signals. Moreover, we retrieve and perform breakdown analysis by utilizing previously unused video category tags provided with the original YouTube videos [2]. The program code of our model and the evaluation approach will be made publicly available.

2. Related Work

2.1. Video Captioning

Early works in video captioning applied *rule-based models* [22, 31, 7], where the idea was to identify a set of video objects and use them to fill predefined templates to generate a sentence. Later, the need for sentence templates was omitted by casting the captioning problem as a machine translation task [37]. Following the success of neural models in translation systems [42], similar methods became widely popular in video captioning [57, 46, 47, 58, 5, 38, 18, 9, 52]. The rationale behind this approach is to train two *Recurrent Neural Networks* (RNNs) in an *encoder-decoder* fashion. Specifically, an encoder inputs a set of video features, accumulates its *hidden state*, which is passed to a decoder for producing a caption.

To further improve the performance of the captioning model, several methods have been proposed, including shared memory between visual and textual domains [49, 34], spatial and temporal attention [56], reinforcement learning [50], semantic tags [11, 32], other modalities [55, 19, 51, 13], and by producing a paragraph instead of one sentence [36, 58].

2.2. Dense Video Captioning

Inspired by the idea of the *dense image captioning* task [20], Krishna *et al*. [24] introduced a problem of dense video captioning and released a new dataset called ActivityNet Captions which leveraged the research in the field [59, 48, 26, 28, 53, 30, 35, 54]. In particular, [48] adopted the idea of the context-awareness [24] and generalized the temporal event proposal module to utilize both past and future contexts as well as an *attentive fusion* to differentiate captions from highly overlapping events. Meanwhile, the concept of *Single Shot Detector* (SSD) [27] was also used to generate event proposals and *reward maximization* for better captioning in [26].

In order to mitigate the intrinsic difficulties of RNNs to model long-term dependencies in a sequence, Zhou *et al*. [59] tailored the recent idea of Transformer [45] for dense video captioning. In [28] the authors noticed that the captioning may benefit from interactions between objects in a video and developed recurrent higher-order interaction module to model these interactions. Xiong *et al*. [53] noticed that many previous models produced redundant captions, and proposed to generate captions in a progressive manner, conditioned on the previous caption while applying paragraph- and sentence-level rewards. Similarly, a "bird-view" correction and two-level reward maximization for a more coherent story-telling have been employed in [30].

Since the human annotation of a video with temporal boundaries and captions for each of them can be laborious, several attempts have been made to address this issue [10, 29]. Specifically, [10] employed the idea of *cycle-consistency* to translate a set of captions to a set of temporal events without any paired annotation, while [29] automatically-collected dataset of an unparalleled-scale exploiting the structure of instructional videos.

The most similar work to our captioning model is [59] that also utilizes a version of the Transformer [45] architecture. However, their model is designed solely for visual features. Instead, we believe that dense video captioning may benefit from information from other modalities.

2.3. Multi-modal Dense Video Captioning

A few attempts has been made to include additional cues like audio and speech [35, 16, 39] for dense video captioning task. Rahman *et al*. [35] utilized the idea of *cycle-consistency* [10] to build a model with visual and audio inputs. However, due to weak supervision, the system did not reach high performance. Hessel *et al*. [16] and Shi *et al*. [39] employ a transformer architecture [45] to encode both video frames and speech segments to generate captions for instructional (cooking) videos. Yet, the high results on a dataset which is restricted to instructional video appear to be not evidential as the speech and the captions are already very close to each other in such videos [29].

In contrast to the mentioned multi-modal dense video captioning methods: (1) we present the importance of the speech and audio modalities on a domain-free dataset, (2) propose a multi-modal dense video captioning module (MDVC) which can be scaled to any number of modalities.

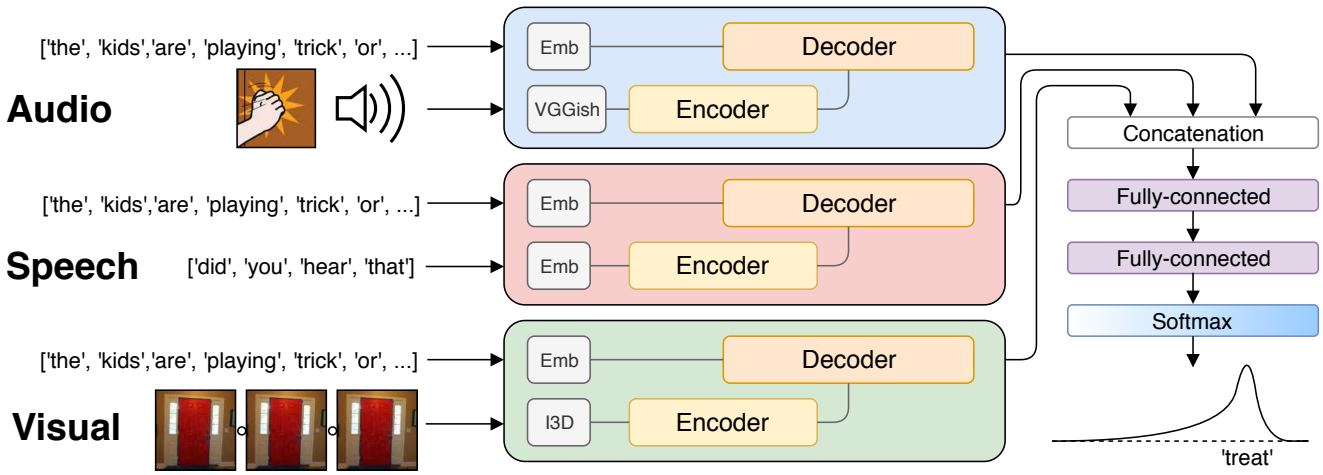

Figure 2. The proposed Multi-modal Dense Video Captioning (MDVC) framework. Given an input consisting of several modalities, namely, audio, speech, and visual, internal representations are produced by a corresponding feature transformer (middle). Then, the features are fused in the multi-modal generator (right) that outputs the distribution over the vocabulary.

3. Proposed Framework

In this section, we briefly outline the workflow of our method referred to as Multi-modal Dense Video Captioning (MDVC) which is shown in Fig. 2. The goal of our method is to temporally localize events on a video and to produce a textual description for each of them. To this end, we apply a two-stage approach.

Firstly, we obtain the temporal event locations. For this task, we employ the *Bidirectional Single-Stream Temporal* action proposals network (Bi-SST) proposed in [48]. Bi-SST applies 3D Convolution network (C3D) [44] to video frames and extracts features that are passed to subsequent *bi-directional LSTM* [17] network. The LSTM accumulates visual cues over time and predicts confidence scores for each location to be start/end point of an event. Finally, a set of event proposals (start/end times) is obtained and passed to the second stage for caption generation.

Secondly, we generate the captions given a proposal. To produce inputs from audio, visual, and speech modalities, we use *Inflated 3D convolutions* (I3D) [6] for visual and *VGGish network* [15] for audio modalities. For speech representation as a text, we employ an external ASR system [1]. To represent the text into a numerical form, we use a similar text embedding which is used for caption encoding. The features are, then, fed to individual transformer models along with the words of a caption from the previous time steps. The output of the transformer is passed into a *generator* which fuses the outputs from all modalities and estimates a probability distribution over the word vocabulary. After sampling the next word, the process is repeated until a special end *token* is obtained. Fig. 1 illustrates an example modality and the corresponding event captions.

3.1. Temporal Event Localization Module

An event localization module is dedicated to generating a set of temporal regions which might contain an event. To achieve this, we employ pre-trained Bidirectional Single-Stream Temporal action proposals network (Bi-SST) proposed in [48] as it has is been shown to reach good performance in the proposal generation task.

Bi-SST inputs a sequence of T RGB frames from a video $V = (x_1, x_2, \ldots, x_F)$ and extracts a set of 4096-d features $V' = (f_1, f_2, \ldots, f_T)$ by applying a 3D Convolution network (C3D) on non-overlapping segments of size 16 with a stride of 64 frames. To reduce the feature dimension, only 500 principal components were selected using PCA.

To account for the video context, events are proposed during forward and backward passes on a video sequence V', and, then, the resulting scores are fused together to obtain the final proposal set. Specifically, during the *forward pass*, LSTM is used to accumulate the visual clues from the "past" context at each position t which is treated as an *ending* point and produce confidence scores for each proposal.

Afterwards, a similar procedure is performed during the *backward pass* where the features V' are used in a reversed order. This empowers the model to have a sense of the "future" context in a video. In contrast to the forward pass, each position is treated as a *starting* point of the proposal. Finally, the confidence scores from both passes are fused by multiplication of corresponding scores for each proposal at each time step, and, then, filtered according to a predefined threshold.

Finally, we obtain a set of N_V event proposals for caption generation $P_V = \{p_j = (\text{start}_j, \text{end}_j, \text{score}_j)\}_{j=1}^{N_V}$.

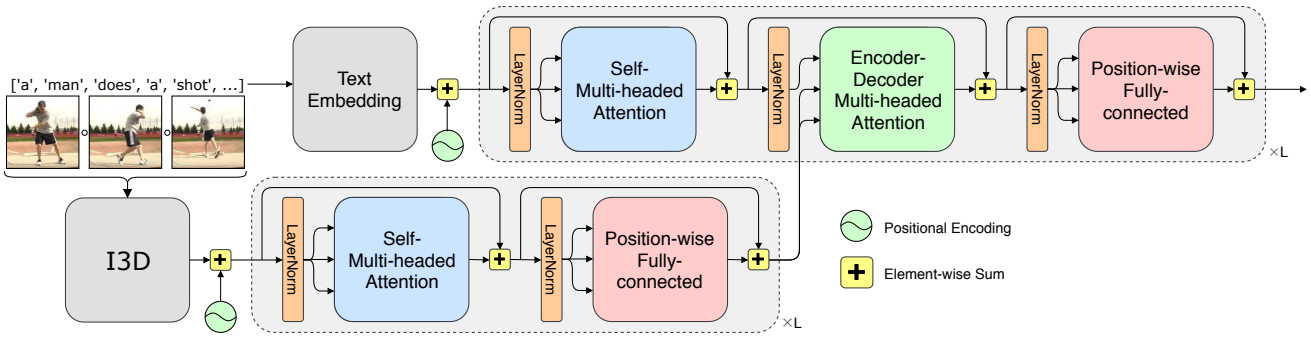

Figure 3. The proposed feature transformation architecture that consists of an encoder (bottom part) and a decoder (top part). The encoder inputs pre-processed and position-encoded features from I3D (in case of the visual modality), and outputs an internal representation. The decoder, in turn, is conditioned on both position-encoded caption that is generated so far and the output of the encoder. Finally, the decoder outputs its internal representation.

3.2. Captioning Module

In this section we explain the captioning based for an example modality, namely, visual. Given a video V and a set of proposals P_V from the event localization module, the task of the captioning module is to provide a caption for each proposal in P_V. In order to extract features from a video V, we employ I3D network [6] pre-trained on the Kinetics dataset which produces 1024-d features. The gap between the extracted features and the generated captions is filled with Transformer [45] architecture which was proven to effectively encode and decode the information in a sequence-to-sequence setting.

3.2.1 Feature Transformer

As shown in Fig. 3, Feature Transformer architecture mainly consists of three blocks: an *encoder*, *decoder*, and *generator*. The encoder inputs a set of extracted features $\mathbf{v}^j = (v_1, v_2, \ldots, v_{T_j})$ temporally corresponding to a proposal p_j from P_V and maps it to a sequence of internal representations $\mathbf{z}^j = (z_1, z_2, \ldots, z_{T_j})$. The decoder is conditioned on the output of the encoder $\mathbf{z}^j$ and the *embedding* $\mathbf{e}^j_{\leqslant t} = (e_1, e_2, \ldots, e_t)$ of the words in a caption $\mathbf{w}^j_{\leqslant t} = (w_1, w_2, \ldots, w_t)$. It produces the representation $\mathbf{g}^j_{\leqslant t} = (g_1, g_2, \ldots, g_t)$ which, in turn, is used by the generator to model a distribution over a vocabulary for the next word $p(w_{t+1}|\mathbf{g}^j_{\leqslant t})$. The next word is selected greedily by obtaining the word with the highest probability until a special *ending token* is sampled. The captioning is initialized with a *starting token*. Both are added to the vocabulary.

Before providing an overview of the encoder, decoder, and generator, we presenting the notion of *multi-headed attention* that acts as an essential part of the decoder and encoder blocks. The concept of the multi-head attention, in turn, heavily relies on *dot-product attention* which we describe next.

Dot-product Attention The idea of the multi-headed attention rests on the *scaled dot-product attention* which calculates the weighted sum of *values*. The weights are obtained by applying the softmax function on the dot-product of each pair of rows of *queries* and *keys* scaled by $\frac{1}{\sqrt{D_k}}$. The scaling is done to prevent the softmax function from being in the small gradient regions [45]. Formally the scaled dot-product attention can be represented as follows

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{D_k}}\right)V, \qquad (1)$$

where Q, K, V are queries, keys, and values, respectively.

Multi-headed Attention The multi-headed attention block is used once in each encoder layer and twice in each decoder layer. The block consists of H *heads* that allows to cooperatively account for information from several representations sub-spaces at every position while preserving the same computation complexity [45]. In a transformer with dimension D_T, each head is defined in the following way

$$\text{head}_h(q, k, v) = \text{Attention}(qW^q_h, kW^k_h, vW^v_h), \qquad (2)$$

where q, k, v are matrices which have D_T columns and the number of rows depending on the position of the multi-headed block, yet with the same number of rows for k and v to make the calculation in (1) to be feasible. The $W^q_h, W^k_h, W^v_h \in \mathbb{R}^{D_T \times D_k}$ are trainable projection matrices that map q, k, v from D_T into $D_k = \frac{D_T}{H}$, asserting D_T is a multiple of H. The multi-head attention, in turn, is the concatenation of all attention heads mapped back into D_T by trainable parameter matrix $W^o \in \mathbb{R}^{D_k \cdot H \times D_T}$:

$$\text{MultiHead}(q, k, v) = \begin{bmatrix} \text{head}_1(q, k, v) \\ \ldots \\ \text{head}_H(q, k, v) \end{bmatrix} W^o. \qquad (3)$$

Encoder The encoder consists of L layers. The first layer inputs a set of features $\mathbf{v}^j$ and outputs an internal representation $\mathbf{z}_1^j \in \mathbb{R}^{T_j \times D_T}$ while each of the next layers treats the output of a previous layer as its input. Each encoder layer l consist of two sub-layers: *multi-headed attention* and *position-wise fully connected network* which are explained later in this section. The input to both sub-layers are normalized using layer normalization [3], each sub-layer is surrounded by a residual connection [14] (see Fig. 3). Formally, the l-th encoder layer has the following definition

$$\overline{\mathbf{z}}_l^j = \text{LayerNorm}(\mathbf{z}_l^j) \tag{4}$$

$$\mathbf{r}_l^j = \mathbf{z}_l^j + \text{MultiHead}(\overline{\mathbf{z}}_l^j, \overline{\mathbf{z}}_l^j, \overline{\mathbf{z}}_l^j) \tag{5}$$

$$\overline{\mathbf{r}}_l^j = \text{LayerNorm}(\mathbf{r}_l^j) \tag{6}$$

$$\mathbf{z}_{l+1}^j = \mathbf{r}_l^j + \text{FCN}(\overline{\mathbf{r}}_l^j), \tag{7}$$

where FCN is the position-wise fully connected network. Note, the multi-headed attention has identical queries, keys, and values ($\overline{\mathbf{z}}_l^j$). Such multi-headed attention block is also referred to as *self*-multi-headed attention. It enables an encoder layer l to account for the information from all states from the previous layer $\mathbf{z}_{l-1}^j$. This property contrasts with the idea of RNN which accumulates only the information from the past positions.

Decoder Similarly to the encoder, the decoder has L layers. At a position t, the decoder inputs a set of embedded words $\mathbf{e}_{\leqslant t}^j$ with the output of the encoder $\mathbf{z}^j$ and sends the output to the next layer which is conditioned on this output and, again, the encoder output $\mathbf{z}^j$. Eventually, the decoder producing its internal representation $\mathbf{g}_{\leqslant t}^j \in \mathbb{R}^{t \times D_T}$. The decoder block is similar to the encoder but has an additional sub-layer that applies multi-headed attention on the encoder output and the output of its previous sub-layer. The decoder employs the layer normalization and residual connections at all three sub-layers in the same fashion as the encoder. Specifically, the l-th decoder layer has the following form:

$$\overline{\mathbf{g}}_l^j = \text{LayerNorm}(\mathbf{g}_{l,\leqslant t}^j) \tag{8}$$

$$\mathbf{b}_l^j = \mathbf{g}_{l,\leqslant t}^j + \text{MultiHead}(\overline{\mathbf{g}}_l^j, \overline{\mathbf{g}}_l^j, \overline{\mathbf{g}}_l^j) \tag{9}$$

$$\overline{\mathbf{b}}_l^j = \text{LayerNorm}(\mathbf{b}_l^j) \tag{10}$$

$$\mathbf{u}_l^j = \mathbf{g}_{l,\leqslant t}^j + \text{MultiHead}(\overline{\mathbf{b}}_l^j, \mathbf{z}^j, \mathbf{z}^j) \tag{11}$$

$$\overline{\mathbf{u}}_l^j = \text{LayerNorm}(\mathbf{u}_l^j) \tag{12}$$

$$\mathbf{g}_{l+1,\leqslant t}^j = \mathbf{u}_l^j + \text{FCN}(\overline{\mathbf{u}}_l^j), \tag{13}$$

where $\mathbf{z}^j$ is the encoder output. Note, similarly to the encoder, (9) is a self-multi-headed attention function while the second multi-headed attention block attends on both the encoder and decoder and is also referred to as *encoder-decoder* attention. This block enables each layer of the decoder to attend all state of the encoder's output $\mathbf{z}^j$.

Position-wise Fully-Connected Network The fully connected network is used in each layer of the encoder and the decoder. It is a simple two-layer neural network that inputs x with the output of the multi-head attention block, and, then, projects each row (or position) of the input x from D_T space onto D_P, $(D_P > D_T)$ and back, formally:

$$\text{FCN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2, \tag{14}$$

where $W_1 \in \mathbb{R}^{D_T \times D_P}$, $W_2 \in \mathbb{R}^{D_P \times D_T}$, and biases b_1, b_2 are trainable parameters, ReLU is a rectified linear unit.

Generator At the position t, the generator consumes the output of the decoder $\mathbf{g}_{\leqslant t}^j$ and produces a distribution over the vocabulary of words $p(w_{t+1}|\mathbf{g}_{\leqslant t}^j)$. To obtain the distribution, the generator applies the softmax function of the output of a fully connected layer with a weight matrix $W_G \in \mathbb{R}^{D_T \times D_V}$ where D_V is a vocabulary size. The word with the highest probability is selected as the next one.

Input Embedding and Positional Encoding Since the representation of textual data is usually sparse due to a large vocabulary, the dimension of the input of a neural language model is reduced with an embedding into a dimension of a different size, namely D_T. Also, following [45], we multiply the embedding weights by $\sqrt{D_T}$. The *position encoding* is required to allow the transformer to have a sense of the order in an input sequence. We adopt the approach proposed for a transformer architecture, *i.e.* we add the output of the combination of sine and cosine functions to the embedded input sequence [45].

3.2.2 Multi-modal Dense Video Captioning

In this section, we present the multi-modal dense video captioning module which, utilises visual, audio, and speech modalities. See Fig. 3 for a schematic representation of the module.

For the sake of speech representation $\mathbf{s}^j = (s_1, s_2, \ldots, s_{T_j^s})$, we use the text embedding of size 512-d that is similar to the one which is employed in the embedding of a caption $\mathbf{w}_{\leqslant t}^j$. To account for the audio information, given a proposal p_j we extract a set of features $\mathbf{a}_j = (a_1, a_2, \ldots, a_{T_j^a})$ applying the 128-d embedding layer of the pre-trained VGGish network [15] on an audio track. While the visual features $\mathbf{v}^j = (v_1, v_2, \ldots v_{T_j^v})$ are encoded with 1024-d vectors by Inflated 3D (I3D) convolutional network [6].

To fuse the features, we create an encoder and a decoder for each modality with dimensions corresponding to the size of the extracted features. The outputs from all decoders are fused inside of the generator, and the distribution of a next word w_{t+1} is formed.

In our experimentation, we found that a simple two-layer fully-connected network applied of a matrix of concatenated features performs the best with the *ReLU* activation after the first layer and the softmax after the second one. Each layer of the network has a matrix of trainable weights: $W_{F_1} \in \mathbb{R}^{D_F \times D_V}$ and $W_{F_2} \in \mathbb{R}^{D_V \times D_V}$ with $D_F = 512 + 128 + 1024$ and D_V is a vocabulary size.

3.3. Model Training

As the training is conducted using mini-batches of size 28, the features in one modality must be of the same length so the features could be stacked into a tensor. In this regard, we *pad* the features and the embedded captions to match the size of the longest sample.

The model is trained by optimizing the Kullback–Leibler divergence loss which measures the "distance" between the ground truth and predicted distributions and averages the values for all words in a batch ignoring the *masked* tokens.

Since many words in the English language may have several synonyms or human annotation may contain mistakes, we undergo the model to be less certain about the predictions and apply Label Smoothing [43] with the smoothing parameter γ on the ground truth labels to mitigate this. In particular, the ground truth distribution over the vocabulary of size D_V, which is usually represented as one-hot encoding vector, the identity is replaced with probability $1 - \gamma$ while the rest of the values are filled with $\frac{\gamma}{D_V - 1}$.

During training, we exploit the *teacher forcing* technique which uses the ground truth sequence up to position t as the input to predict the next word instead of using the sequence of predictions. As we input the whole ground truth sequence at once and predicting the next words at each position, we need to prevent the transformer from peeping for the information from the next positions as it attends to all positions of the input. To mitigate this, we apply masking inside of the self-multi-headed attention block in the decoder for each position higher than $t - 1$, following [45].

The details on the feature extraction and other implementation details are available in the supplementary materials.

4. Experiments

4.1. Dataset

We perform our experiments using ActivityNet Captions dataset [24] that is considered as the standard benchmark for dense video captioning task. The dataset contains approximately 20k videos from YouTube and split into 50/25/25 % parts for training, validation, and testing, respectively. Each video, on average, contains 3.65 temporally localized captions, around 13.65 words each, and two minutes long. In addition, each video in the validation set is annotated twice by different annotators. We report all results using the validation set (no ground truth is provided for the test set).

Method	GT Proposals			Learned Proposals		
	B@3	B@4	M	B@3	B@4	M
Seen full dataset						
Krishna *et al*. [24]	4.09	1.60	8.88	1.90	0.71	5.69
Wang *et al*. [48]	–	–	10.89	2.55	**1.31**	5.86
Zhou *et al*. [59]	5.76	2.71	11.16	2.42	1.15	4.98
Li *et al*. [26]	4.55	1.62	10.33	2.27	0.73	6.93
Seen part of the dataset						
Rahman *et al*. [35]	3.04	1.46	7.23	1.85	0.90	4.93
MDVC	4.12	1.81	10.09	2.31	0.92	6.80
MDVC, no missings	**5.83**	**2.86**	**11.72**	**2.60**	1.07	**7.31**

Table 1. The results of the dense video captioning task on the ActivityNet Captions validation sets in terms of BLEU–3,4 (B@3, B@4) and METEOR (M). The related methods are compared with the proposed approach (MDVC) in two settings: on the full validation dataset and a part of it with the videos with all modalities present for a fair comparison ("no missings"). Methods are additionally split into the ones which "saw" all training videos and another ones which trained on partially available data. The results are presented for both ground truth (GT) and learned proposals.

The dataset itself is distributed as a collection of links to YouTube videos, some of which are no longer available. Authors provide pre-computed C3D features and frames at 5fps, but these are not suitable for our experiments. At the time of writing, we found 9,167 (out of 10,009) training and 4,483 (out of 4,917) validation videos which is, roughly, 91 % of the dataset. Out of these 2,798 training and 1,374 validation videos (approx. 28 %) contain at least one speech segment. The speech content was obtained from the *closed captions* (CC) provided by the YouTube ASR system which can be though as subtitles.

4.2. Metrics

We are evaluating the performance of our model using BLEU@N [33] and METEOR [8]. We regard the METEOR as our primary metric as it has been shown to be highly correlated with human judgement in a situation with a limited number of references (only one, in our case).

We employ the official evaluation script provided in [23]. Thus, the metrics are calculated if a proposed event and a ground truth location of a caption overlaps more than a specified *temporal Intersection over Union* (tIoU) and zero otherwise. All metric values are averaged for every video, and, then, for every threshold tIoU in $[0.3, 0.5, 0.7, 0.9]$. On the validation, we average the resulting scores for both validation sets. For the learned proposal setting, we report our results on at most 100 proposals per video.

Notably, up to early 2017, the evaluation code had an issue which previously overestimated the performance of the algorithms in the learned proposal setting [30]. Therefore, we report the results using the new evaluation code.

Model	Params. ($\times 10^6$)	Metric B@4	M
Feature Transf. (random)	42	0.88	7.16
Bi-GRU	55	1.44	9.47
Feature Transformer	42	**1.84**	**9.62**

Table 2. Comparison of the Feature Transformer and the Bi-directional GRU (Bi-GRU) architectures in terms of BLEU-4 (B@4), METEOR (M), and a number of model parameters. The input to all models is visual modality (I3D). The results indicate the superior performance of the Feature Transformer on all metrics. Additionally, we report the random input baseline which acts as a lower performance bound. The best results are highlighted.

Modality V	A	S	Fusion	Params. ($\times 10^6$)	Metric B@4	M
✔			–	42	1.61	9.64
	✔		–	5	1.03	8.01
✔	✔		Average probs.	46	1.68	9.71
✔	✔		Concat. + 2 FC	149	1.73	9.87
✔			No, 2 FC	145	1.62	9.69
✔	✔	✔	Concat. + 2 FC	179	**1.81**	**10.09**

Table 3. The performance of the proposed MDVC framework with different input modalities (V-visual, A-audio, S-speech) and feature fusion approaches: probability averaging and concatenation of two fully-connected layers (Concat. + 2 FC). Also, we report the comparison between audio-visual MDVC with visual-only MDVC with similar model capacities (2 FC).

4.3. Comparison with Baseline Methods

We compare our method with five related approaches, namely Krishna *et al*. [24], Wang *et al*. [48], Zhou *et al*. [59], Li *et al*. [26], and Rahman *et al*. [35]. We take the performance values from the original papers, except for [26], and [59], which are taken from [30] due to the evaluation issue (see Sec. 4.2).

The lack of access to the full ActivityNet Captions dataset makes strictly fair comparison difficult as we have less training and validation videos. Nevertheless, we present our results in two set-ups: 1) full validation set with random input features for missing entries, and 2) videos with all three modalities present (video, audio, and speech). The first one is chosen to indicate the lower bound of our performance with the full dataset. Whereas, the second one (referred to as "*no missings*") concentrates on the multi-modal setup, which is the main contribution of our work.

The obtained results are presented in Tab. 1. Our method (MDVC) achieves comparable or better performance, even though we have access to smaller training set and 9 % of the validation videos are missing (replaced with random input features). Furthermore, if all three modalities are present, our method outperforms all baseline approaches in the case of both GT and learned proposals. Notably, we outperform [59] which is also based on the transformer architecture and account for the optical flow. This shows the superior performance of our captioning module which, yet, trained on the smaller amount of data.

4.4. Ablation Studies

In this section, we perform an ablation analysis highlighting the effect of different design choices of our method. For all experiments, we use the full unfiltered ActivityNet Captions validation set with ground truth event proposals.

Firstly, we assess the selection of the model architecture. To this end, we implemented a version of our method where the transformer was replaced by *Bidirectional Recurrent Neural Network with Gated Recurrent Units with attention* (Bi-GRU), proposed in [4]. To distil the effect of the change in architecture, the results are shown for visual-only models. Both Bi-GRU and the transformer input I3D features extracted from 64 RGB and optical flow frames (the final model inputs 24 frames). Finally, we set a lower bound for the feature performance by training a transformer model with random video features. Tab. 2 shows the comparison. To conclude, we observe that the feature transformer-based model is not only uses less parameters but also achieves better performance in dense video captioning task. Moreover, both method clearly surpasses the random baseline.

Secondly, we evaluate the contribution of different modalities in our framework. Tab. 3 contains the results for different modality configurations as well as for two feature fusion approaches. Specifically, averaging of the output probabilities and concatenation of the outputs of all modalities and applying two fully connected (FC) layers on top. We observe that audio-only model has the worst performance, followed by the visual only model, and the combination of these two. Moreover, the concatenation and FC layers result in better performance than averaging. To further assess if the performance gain is due to the additional modalities or to the extra capacity in the FC layers, we trained a visual-only model with two additional FC layers. The results indicate that such configuration performs worse than any bi-modal setup. Overall, we conclude that the final model with all three modalities performs best among all tested set-ups, which highlights the importance of multi-modal setting in dense video captioning task.

Fig. 4 shows a qualitative comparison between different models in our ablation study. Moreover, we provide the corresponding captions from the best performing baseline method (Zhuo *et al*. [59]). We noticed the following pattern: the audio-modality produces coherent sentences and captures the concepts of speaking in the video. However, there are clear mistakes in the caption content. In contrast, the model with all three modalities manages to capture

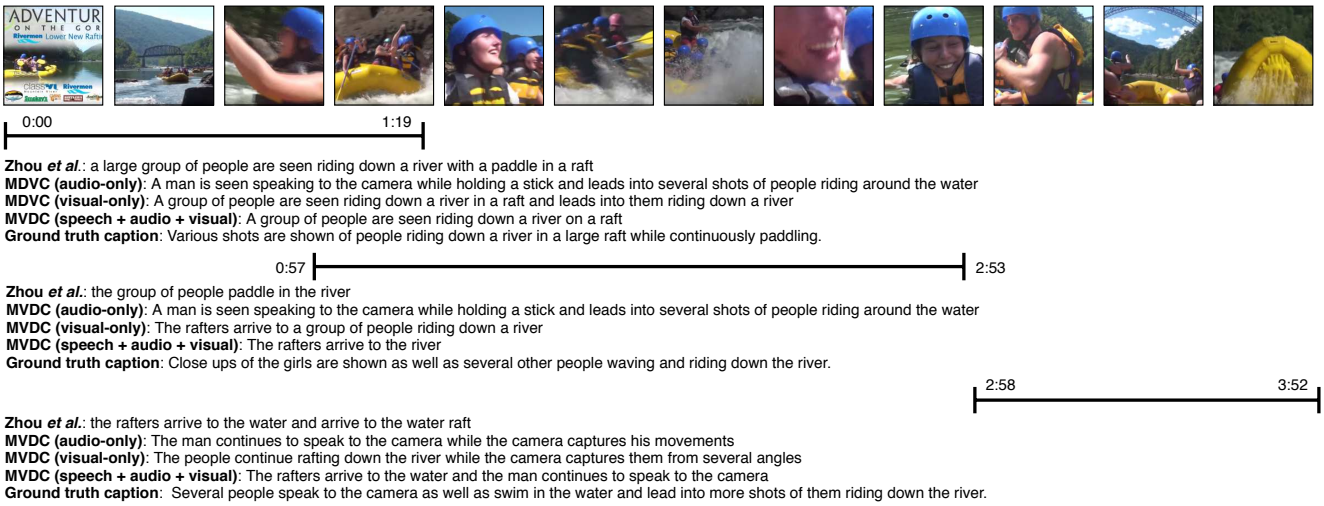

Figure 4. The qualitative captioning results for an example video from the ActivityNet Captions validation set. In the video, the speaker describes the advantages of rafting on this particular river and their club. Occasionally, people are shown rapturously speaking about how fun it is. Models that account for audio modality tend to grasp the details of the speaking on the scene while the visual-only models fail at this. We invite the reader to watch the example YouTube video for a better impression (`xs5imfBbWmw`).

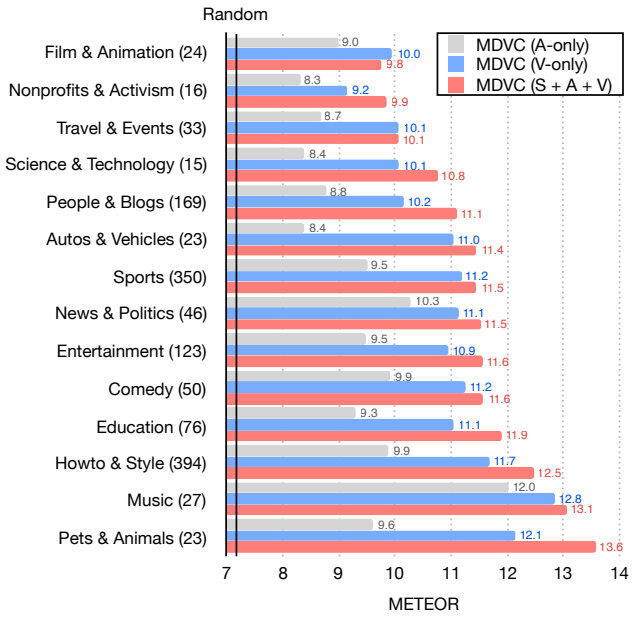

Figure 5. The results are split for category and version of MDVC. The number of samples per category is given in parenthesis. The METEOR axis is cut up to the random performance level (7.16).

the man who speaks to the camera which is also present in the ground truth. Both visual-only MDVC and Zhuo *et al.* struggle to describe the audio details.

Finally, to test whether our model improves the performance in general rather than in a specific video category, we report the comparison of the different versions of MDVC per category. To this end, we retrieve the category labels from the YouTubeAPI [2] (US region) for every available ActivityNet Captions validation video. These labels are given by the user when uploading the video and roughly represent the video content type. The comparison is shown in Fig. 5. The results imply a consistent gain in performance within each category except for categories: "Film & Animation" and "Travel & Events" which might be explained by the lack of correspondence between visual and audio tracks. Specifically, the video might be accompanied by music, *e.g.* promotion of a resort. Also, "Film & Animation" contains cartoon-like movies which might have a realistic soundtrack while the visual track is goofy.

5. Conclusion

The use of different modalities in computer vision is still an underrepresented topic and, we believe, deserves more attention. In this work, we introduced a multi-modal dense video captioning module (MDVC) and shown the importance of the audio and speech modalities for dense video captioning task. Specifically, MDVC is based on the transformer architecture which encodes the feature representation of each modality for a specific event proposal and produces a caption using the information from these modalities. The experimentation, conducted employing the ActivityNet Captions dataset, shows the superior performance of a captioning module to the visual-only models in the existing literature. Extensive ablation study verifies this conclusion. We believe that our results firmly indicate that future works in video captioning should utilize a multi-modal input.

Acknowledgments Funding for this research was provided by the Academy of Finland projects 327910 & 324346. The authors acknowledge CSC — IT Center for Science, Finland, for computational resources.

References

[1] YouTube Data API Video Captions. https://developers.google.com/youtube/v3/docs/captions, [Accessed 1 November 2019].

[2] YouTube Data API Video Categories. https://developers.google.com/youtube/v3/docs/videoCategories, [Accessed 1-November-2019].

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[5] L. Baraldi, C. Grana, and R. Cucchiara. Hierarchical boundary-aware neural encoder for video captioning. In *CVPR*, 2017.

[6] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

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

[8] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *ACL Workshop*, 2014.

[9] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *TPAMI*, 2017.

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

[11] Z. Gan, C. Gan, X. He, Y. Pu, K. Tran, J. Gao, L. Carin, and L. Deng. Semantic compositional networks for visual captioning. In *CVPR*, 2017.

[12] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *ICASSP*, 2017.

[13] Wang-Li Hao, Zhaoxiang Zhang, and He Guan. Integrating both visual and audio cues for enhanced video caption. In *AAAI*, 2017.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[15] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron Weiss, and Kevin Wilson. CNN architectures for large-scale audio classification. In *ICASSP*, 2017.

[16] Jack Hessel, Bo Pang, Zhenhai Zhu, and Radu Soricut. A case study on combining ASR and visual features for generating instructional video captions. In *CoNLL*, 2019.

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.

[18] C. Hori, T. Hori, T. Lee, Z. Zhang, B. Harsham, J. R. Hershey, T. K. Marks, and K. Sumi. Attention-based multimodal fusion for video description. In *ICCV*, 2017.

[19] Chiori Hori, Takaaki Hori, Gordon Wichern, Jue Wang, Teng-Yok Lee, Anoop Cherian, and Tim K. Marks. Multimodal attention for fusion of audio and spatiotemporal features for video description. In *CVPR Workshops*, 2018.

[20] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *CVPR*, 2016.

[21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[22] Atsuhiro Kojima, Takeshi Tamura, and Kunio Fukunaga. Natural language description of human activities from video images based on concept hierarchy of actions. *IJCV*, 2002.

[23] Ranjay Krishna. Evaluation code for dense-captioning events in videos. https://github.com/ranjaykrishna/densevid_eval/tree/9d4045aced3d827834a5d2da3c9f0692e3f33c1c.

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

[25] Yong Jae Lee, Joydeep Ghosh, and Kristen Grauman. Discovering important people and objects for egocentric video summarization. In *CVPR*, 2012.

[26] Yehao Li, Ting Yao, Yingwei Pan, Hongyang Chao, and Tao Mei. Jointly localizing and describing events for dense video captioning. In *CVPR*, 2018.

[27] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *ECCV*, 2016.

[28] C. Ma, A. Kadav, I. Melvin, Z. Kira, G. AlRegib, and H. P. Graf. Attend and interact: Higher-order object interactions for video understanding. In *CVPR*, 2018.

[29] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019.

[30] Jonghwan Mun, Linjie Yang, Zhou Ren, Ning Xu, and Bohyung Han. Streamlined dense video captioning. In *CVPR*, 2019.

[31] Mun Wai Lee, A. Hakeem, N. Haering, and Song-Chun Zhu. Save: A framework for semantic annotation of visual events. In *CVPR Workshops*, 2008.

[32] Y. Pan, T. Yao, H. Li, and T. Mei. Video captioning with transferred semantic attributes. In *CVPR*, 2017.

[33] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *ACL*, 2002.

[34] Wenjie Pei, Jiyuan Zhang, Xiangrong Wang, Lei Ke, Xiaoyong Shen, and Yu-Wing Tai. Memory-attended recurrent network for video captioning. In *CVPR*, 2019.

[35] Tanzila Rahman, Bicheng Xu, and Leonid Sigal. Watch, listen and tell: Multi-modal weakly supervised dense event captioning. In *ICCV*, 2019.

[36] Anna Rohrbach, Marcus Rohrbach, Wei Qiu, Annemarie Friedrich, Manfred Pinkal, and Bernt Schiele. Coherent multi-sentence video description with variable level of detail.

In Xiaoyi Jiang, Joachim Hornegger, and Reinhard Koch, editors, *Pattern Recognition*. Springer International Publishing, 2014.

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

[38] Z. Shen, J. Li, Z. Su, M. Li, Y. Chen, Y. Jiang, and X. Xue. Weakly supervised dense video captioning. In *CVPR*, 2017.

[39] Botian Shi, Lei Ji, Yaobo Liang, Nan Duan, Peng Chen, Zhendong Niu, and Ming Zhou. Dense procedure captioning in narrated instructional videos. In *ACL*, 2019.

[40] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[41] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018.

[42] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NeurIPS*, 2014.

[43] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[44] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In *ICCV*, 2015.

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

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

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

[48] J. Wang, W. Jiang, L. Ma, W. Liu, and Y. Xu. Bidirectional attentive fusion with context gating for dense video captioning. In *CVPR*, 2018.

[49] J. Wang, W. Wang, Y. Huang, L. Wang, and T. Tan. M3: Multimodal memory modelling for video captioning. In *CVPR*, 2018.

[50] X. Wang, W. Chen, J. Wu, Y. Wang, and W. Y. Wang. Video captioning via hierarchical reinforcement learning. In *CVPR*, 2018.

[51] Xin Wang, Yuan-Fang Wang, and William Yang Wang. Watch, listen, and describe: Globally and locally aligned cross-modal attentions for video captioning. In *NAACL-HLT*, 2018.

[52] Xian Wu, Guanbin Li, Qingxing Cao, Qingge Ji, and Liang Lin. Interpretable video captioning via trajectory structured localization. In *CVPR*, 2018.

[53] Yilei Xiong, Bo Dai, and Dahua Lin. Move forward and tell: A progressive generator of video descriptions. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *ECCV*, 2018.

[54] H. Xu, B. Li, V. Ramanishka, L. Sigal, and K. Saenko. Joint event detection and description in continuous video streams. In *WACV*, 2019.

[55] Jun Xu, Ting Yao, Yongdong Zhang, and Tao Mei. Learning multimodal attention lstm networks for video captioning. In *ACM*, 2017.

[56] C. Yan, Y. Tu, X. Wang, Y. Zhang, X. Hao, Y. Zhang, and Q. Dai. STAT: Spatial-temporal attention mechanism for video captioning. *Transactions on Multimedia*, 2020.

[57] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *ICCV*, 2015.

[58] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *CVPR*, 2016.

[59] Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *CVPR*, 2018.