A Better Use of Audio-Visual Cues: Dense Video Captioning with Bi-modal Transformer

Vladimir Iashin
vladimir.iashin@tuni.fi
Esa Rahtu
esa.rahtu@tuni.fi

Computing Sciences
Tampere University
Tampere, Finland

Abstract

Dense video captioning aims to localize and describe important events in untrimmed videos. Existing methods mainly tackle this task by exploiting only visual features, while completely neglecting the audio track. Only a few prior works have utilized both modalities, yet they show poor results or demonstrate the importance on a dataset with a specific domain. In this paper, we introduce *Bi-modal Transformer* which generalizes the Transformer architecture for a bi-modal input. We show the effectiveness of the proposed model with audio and visual modalities on the dense video captioning task, yet the module is capable of digesting any two modalities in a sequence-to-sequence task. We also show that the pre-trained bi-modal encoder as a part of the bi-modal transformer can be used as a feature extractor for a simple proposal generation module. The performance is demonstrated on a challenging *ActivityNet Captions* dataset where our model achieves outstanding performance. The code is available: `v-iashin.github.io/bmt`

1 Introduction

Current video sharing platforms contain a large amount of video material. The ability to generate descriptions of this content would be highly valuable for many tasks, such as content-based retrieval or recommendation [25, 44]. Moreover, they would enable visually-impaired people to consume video material and improve their quality of life [38].

This kind of video descriptions are usually provided as natural language sentences or *captions*, a compact and intuitive format and, most importantly, can be digested by humans. Early works [46, 47, 56, 58] described the video content with only one sentence, which might be too "sparse" for long videos – one might try to think up a relatively short sentence which describes the whole film. To mitigate this issue, [20] proposed *dense video captioning* which requires a model to, first, localize "events", and, then, to produce one-sentence description for each of them instead of generating one caption for the entire film (see Fig. 1).

The task is usually formulated as a *sequence-to-sequence* (video to caption) task. Therefore, the progress in the field is significantly influenced by advances in machine translation. Hence, many models rely on an encoder-decoder architecture which consists of two *recurrent neural networks* (RNNs) or, recently-proposed *Transformer*-like model [45]. An event

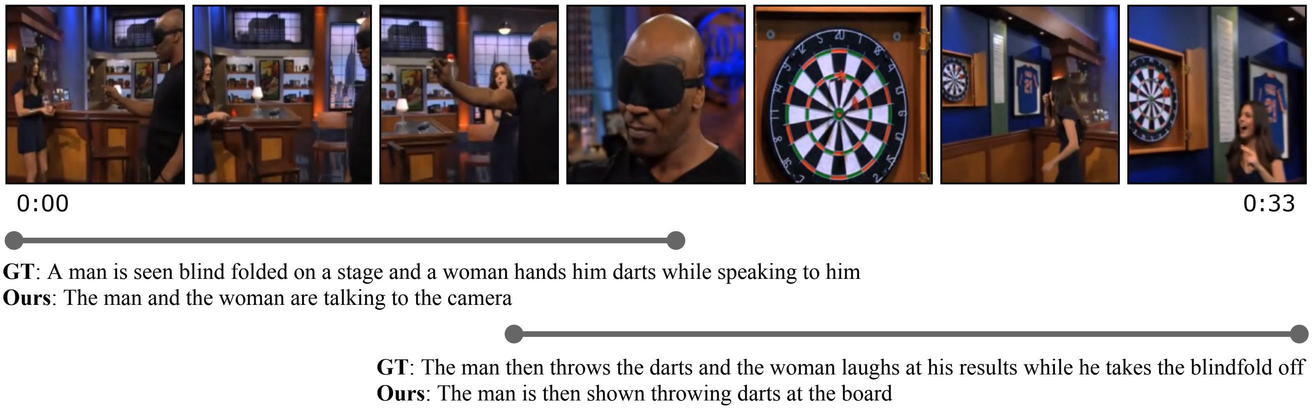

Figure 1: Example video with the predictions of our model alongside the ground truth.

localization module usually utilizes an RNN structure which first encodes the input to produce a hidden representation and, then, makes predictions using this representation.

Considering the natural co-occurrence of visual and audio tracks in a video and the fact that human perception is multi-modal, recent advances in deep learning practice audio-visual training [24, 27, 59, 63, 64]. Yet, most of the existing works on dense video captioning employ only visual inputs. In this work, we address this issue by introducing a novel bi-modal transformer with the multi-headed proposal generator. Our captioning module is inspired by the transformer architecture and, more precisely, how the attention module fuses the information from both sequences. While an efficient object detector *YOLO* [35] inspires the design of each proposal head in the bi-modal multi-headed proposal generator.

The proposed method effectively utilizes audio and visual cues. We demonstrate the performance of our model on the challenging open-domain ActivityNet Captions dataset [20]. The results show the state-of-the-art performance of our bi-modal dense video captioning module as well as our bi-modal proposal generator on BLEU@3–4 and F1 metrics.

2 Related Work

The dense video captioning task requires a model to, first, localize events within a video and, then, to produce a textual one-sentence description of what is happening during the event. The dense video captioning task branches out from the *video captioning* which task is to caption a video without localizing the event. The video captioning field evolved from hand-crafted rule models [6, 19, 21] to *encoder-decoder* architectures [46, 47, 56, 58] inspired by advances in machine translation [39]. Later, the captioning models were further enhanced by *semantic tagging* [11, 28], *reinforcement learning* [51], *attention* [55], *extended memory* [31, 50], and other modalities [13, 16, 52, 54].

2.1 Dense Video Captioning

The task of dense video captioning, as well as a test-bed, ActivityNet Captions dataset, were introduced by Krishna *et al*. [20] who utilized the idea of the *Deep Action Proposals* network [10] to generate event proposals and an LSTM network to encode the context and generate captions. The idea of context-awareness was further developed in [49] who employed a bi-directional variant of *Single-Stream Temporal* Action proposal network (SST) [3] which makes better use of the video context, an LSTM network with *attentive fusion and context* *gating* was used to generate context-aware captions. Zhou *et al*. [62] adapted *Transformer* architecture [45] to tackle the task and used transformer *encoder*'s output as input to a modification of *ProcNets* [61] to generate proposals.

Recently, the idea of reinforcement learning was found to be beneficial for image captioning (*Self-critical Sequence Training* (SCST)) [37] and, hence, applied in dense video captioning as well. More precisely, the SCST was used in a captioning module to optimize the non-differentiable target metric, *e.g*. METEOR [7]. Specifically, Li *et al*. [22] integrated the reward system and enriched *Single-Shot-Detector*-like structure [23] with descriptiveness regression for proposal generation. Similarly, Xiong *et al*. [53] used an LSTM network trained with the sentence- and paragraph-level rewards for maintaining coherent and concise story-telling, while the event proposal module was adopted from *Structured Segment Networks* [60]. Mun *et al*. [26] further developed the idea of coherent captioning by observing the overall context and optimizing two-level rewards, an SST module is used for proposal generation, and a *Pointer Network* [48] to distill proposal candidates.

Another direction of research relies on weak supervision which is designed to mitigate the problem of laborious annotation of the datasets. To this end, Duan *et al*. [9] proposed an *autoencoder* architecture which generates proposals and, then, captions them while being supervised only with a set of non-localized captions in a *cycle-consistency* manner. However, the results appeared to be far from the supervised methods.

2.2 Multi-modal Dense Video Captioning

It is natural to assume that, besides visual information, a video understanding system might benefit from the cues contained in other modalities like audio [33], speech (subtitles) [40], or both [17]. Specifically, Rahman *et al*. [33] were the first to include audio modality into the dense video captioning set up. They borrowed the idea of cycle-consistency from [9] and employed *multi-modal Tucker decomposition* [2] to combine information from both modalities and pass it to a *GRU*-based [5] caption decoder. However, since the model is trained in a weakly supervised setting, the results do not reach the performance of the supervised models.

Shi *et al*. [40] proposed to utilize the corresponding speech along with frame features to further improve captioning performance on cooking videos. They suggested employing a transformer's encoder to encode video frames and subtitle *embeddings* produced by a pre-trained *BERT* model [8]. Next, an LSTM generates proposals, and the other two LSTMs were used for the encoder-decoder captioning module. Despite the significant gains in captioning performance, we believe these findings are not conclusive as instructional videos is an ill-suited domain to show the benefits of the speech modality for a captioning task since subtitles alone can be a very accurate proxy for captions in such videos (see [25]).

In contrast, Iashin *et al*. [17] showed the importance of the speech modality on a free-domain dataset. They proposed to train three transformers for each modality individually and fuse features by concatenation before predicting the next caption word while borrowing the proposal generator from [49]. However, the suggested approach for feature fusion is rather straightforward and inefficient. Moreover, the adopted proposal generator is based solely on video features which contrasts with the idea of the dense video captioning task.

Our method is mostly similar to [17], yet we show significantly better results on the task while utilizing only visual and audio cues. Besides, our proposal generator does employ both modalities and significantly outperforms the state-of-the-art. Furthermore, we present a single model which utilizes bi-modal encoder for both: the proposal generator and captioning module, making it an elegant approach for the dense video captioning task.

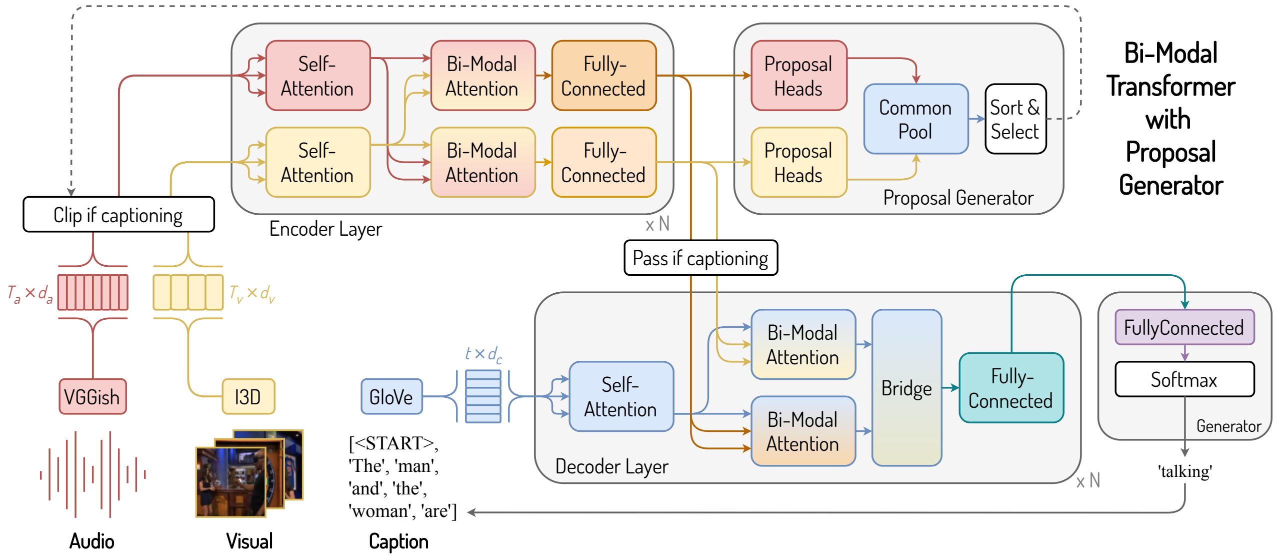

Figure 2: The design of Bi-modal Transformer with Multi-headed Proposal Generator. The proposed model inputs features extracted by VGGish, I3D, and GloVe pre-trained models (bottom left). Then, the bi-modal encoder with N layers processes the audio and visual features and passes its bi-modal representation to the proposal generator (top). After, the generated proposals are used to clip the input features (left). The clipped features are passed through the encoder again. The output of the encoder, then, is used at every layer (N) of the bi-modal decoder (bottom). The decoder attends to the bi-modal encoder's representation as well as the previous caption words and produces its internal representation of the context. This representation is passed to the generator (right) to generate the next word. Residual connections are removed for clarity. Best viewed in color.

3 Our Framework

Our approach consists of two parts: the *bi-modal transformer* and *multi-headed proposal generator* (see Fig. 2). The model expects the input to be a set of continuous features stacked together in a sequence. To represent a visual stream, we use a pre-trained *Inflated 3D* (I3D) network [4] while for the audio stream we employ pre-trained *VGGish* [15], the tokens (roughly, words) are embedded with pre-trained *GloVe* [32] (see Sec. 6.2 for implementation details). Also, since the transformer is *permutation invariant* it has no sense of recurrence. Thus, the order of features within a sequence is preserved by adding the *positional encoding* to the output of the embedding layers. Following [45], we use *cosine* and *sine* functions.

Next, the audio and visual sequences, are passed through the transformer's bi-modal N-layered *encoder* to produce bi-modal sequence representations utilizing novel *bi-modal multi-headed attention* blocks to fuse the features from both sequences. Then, the novel proposal generator utilizes these features to generate proposals and their confidence scores. After, a pre-defined number of most confident proposals are selected to clip the input feature sequences. Next, the clipped features are processed with the encoder to re-represent the features considering only the features which are left after clipping.

The bi-modal encoder's representation is used at every layer in the bi-modal *decoder*. Concretely, the encoder's outputs are passed to the corresponding bi-modal attention blocks in the decoder layer along with the representation of the previously generated caption words. The last-layer representation of the decoder is used in the *generator* where the next caption word is produced. To avoid an empty input to the decoder in the beginning, a special *start-token* is used. The caption is generated word-by-word until a special *end-token* is sampled.

This section, first, presents the design of the captioning module (Sec. 3.1) and, second, the proposal generator (Sec. 3.2) while the training procedure is explained in Sec. 3.3.

3.1 Captioning Module

The task of dense video captioning requires to produce a caption for each proposal. Therefore, *bi-modal encoder* inputs audio A and visual V feature sequences which temporally correspond to the proposal and outputs two sequences: audio-attended visual features V^a and visual-attended audio features A^v. These features are used by the *bi-modal decoder* which attends to these features and the previous caption words $(c_1, c_2, \dots, c_t)$. Finally, the bi-modal decoder outputs the representation which is employed to model a distribution of the next caption word (c_{t+1}) over the vocabulary. The proposal index is omitted for clarity.

Bi-modal Encoder In contrast to the encoder in [45], our bi-modal encoder inputs two streams: audio ($A \in \mathrm{R}^{T_a \times d_a}$) and visual ($V \in \mathrm{R}^{T_v \times d_v}$) features corresponding to the proposal. Then, the features are passed in a stack of N encoder layers. Instead of two, each layer has three sub-layers: *self-attention*, *bi-modal attention* (new), and *position-wise fully-connected* layers. Specifically, given $A_0^{\text{fc}} = A$ and $V_0^{\text{fc}} = V$, an n^{th} encoder layer is defined as

$$A_n^{\text{self}} = \text{MultiHeadAttention}(A_{n-1}^{\text{fc}}, A_{n-1}^{\text{fc}}, A_{n-1}^{\text{fc}}), \quad \text{// audio self-attention} \quad (1)$$

$$V_n^{\text{self}} = \text{MultiHeadAttention}(V_{n-1}^{\text{fc}}, V_{n-1}^{\text{fc}}, V_{n-1}^{\text{fc}}), \quad \text{// visual self-attention} \quad (2)$$

$$A_n^{\text{mm}} = \text{MultiHeadAttention}(A_n^{\text{self}}, V_n^{\text{self}}, V_n^{\text{self}}), \quad \text{// visual-attended audio feats.} \quad (3)$$

$$V_n^{\text{mm}} = \text{MultiHeadAttention}(V_n^{\text{self}}, A_n^{\text{self}}, A_n^{\text{self}}), \quad \text{// audio-attended visual feats.} \quad (4)$$

$$A_n^{\text{fc}} = \text{TwoFullyConnected}(A_n^{\text{mm}}), \quad \text{// } \mathrm{R}^{T_a \times d_a} \leftarrow \mathrm{R}^{T_a \times 4d_a} \leftarrow \mathrm{R}^{T_a \times d_a} \quad (5)$$

$$V_n^{\text{fc}} = \text{TwoFullyConnected}(V_n^{\text{mm}}), \quad \text{// } \mathrm{R}^{T_v \times d_v} \leftarrow \mathrm{R}^{T_v \times 4d_v} \leftarrow \mathrm{R}^{T_v \times d_v} \quad (6)$$

where all sub-layers have distinct sets of trainable weights and mostly resemble the blocks of Transformer [45], yet we allow the dimension of the weights in multi-headed attention in (3) & (4) to be different for both modalities because we expect them to have a different size. We define the multi-headed attention in Sec. 6.1. The encoder outputs visual-attended audio features ($A^v = A_N^{\text{fc}}$) and audio-attended visual features ($V^a = V_N^{\text{fc}}$), which are used the decoder.

Bi-modal Decoder The bi-modal decoder inputs the previous sequence of caption words $C_t = (c_1, c_2, \dots, c_t) \in \mathrm{R}^{t \times d_c}$ and, opposed to the original Transformer's decoder [45], ours gets the output from the bi-modal encoder ($A^v \in \mathrm{R}^{T_a \times d_a}$, $V^a \in \mathrm{R}^{T_v \times d_v}$). Thus, instead of three, it has four sub-layers: self-attention, *bi-modal* encoder-decoder attention (new), *bridge* (new), & position-wise fully-connected layers. For $C_0^{\text{fc}} = C_t$, an n^{th} decoder layer is defined as

$$C_n^{\text{self}} = \text{MultiHeadAttention}(C_{n-1}^{\text{fc}}, C_{n-1}^{\text{fc}}, C_{n-1}^{\text{fc}}), \quad \text{// caption self-attention} \quad (7)$$

$$C_n^{A^v} = \text{MultiHeadAttention}(C_n^{\text{self}}, A^v, A^v), \quad \text{// audio-visual attended prev. caps.} \quad (8)$$

$$C_n^{V^a} = \text{MultiHeadAttention}(C_n^{\text{self}}, V^a, V^a), \quad \text{// visual-audio attended prev. caps.} \quad (9)$$

$$C_n^{\text{mm}} = \text{OneFullyConnected}\big([C_n^{A^v}, C_n^{V^a}]\big), \quad \text{// } \mathrm{R}^{t \times d_c} \leftarrow \mathrm{R}^{t \times 2d_c}; [\cdot,\cdot] \text{ — concat.} \quad (10)$$

$$C_n^{\text{fc}} = \text{TwoFullyConnected}(C_n^{\text{mm}}), \quad \text{// } \mathrm{R}^{t \times d_c} \leftarrow \mathrm{R}^{t \times 4d_c} \leftarrow \mathrm{R}^{t \times d_c} \quad (11)$$

where, as in the encoder, trainable weights have distinct dimensions depending on a modality and are not shared across sub-layers. The decoder outputs caption features ($C_t^{\text{av}} = C_N^{\text{fc}}$).

Generator The purpose of the generator is to model the distribution for the next caption word c_{t+1} given the output of the decoder $C_t^{\text{av}} \in \mathrm{R}^{t \times d_c}$. Therefore, the generator is, usually, a fully-connected layer with the softmax activation which maps the caption features of size d_c into a dimension corresponding to the size of the vocabulary in the training set.

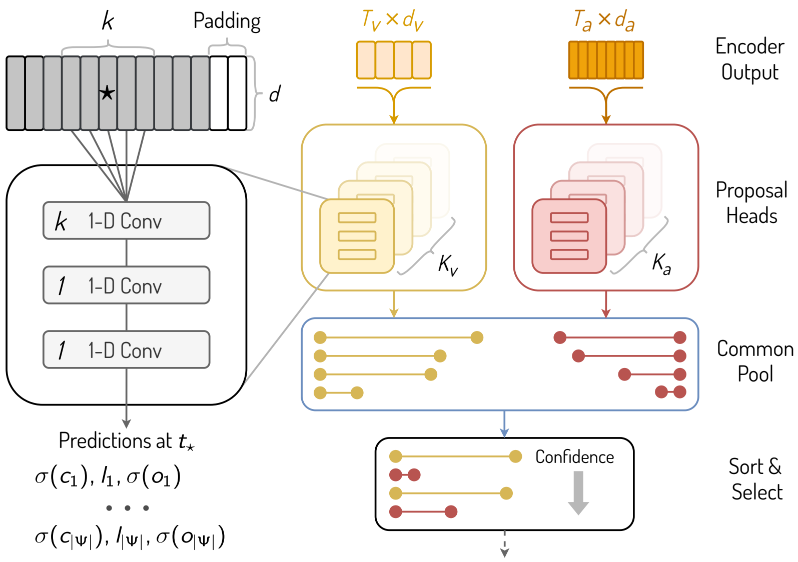

Figure 3: The Bi-modal Multi-headed Proposal Generator inputs the two-stream output from the bi-modal encoder, processes it with two stacks of proposal generation heads. The predictions from all heads form a common pool of predictions. Thus, the pool consists of $T_v \cdot K_v \cdot |\Psi_v| + T_a \cdot K_a \cdot |\Psi_a|$ proposals, which are sorted on the confidence score and passed back to clip input features to the captioning module.

Residual Connection Following the original Transformer architecture, we employ the *residual connection* [14] surrounding each sub-layer of the encoder and decoder except for the bridge layer since in- and out-dimensions are different. Additionally, we adopt Layer Normalization [1] before applying a sub-layer: $x + \text{sub-layer}\big(\text{LayerNorm}(x)\big)$.

Dropout We also regularize our model with *dropout* [41] which is applied: a) before adding the residual in the residual connection, b) before the activation in the bridge layer, c) on outputs of the positional encoding, d) between layers in the position-wise fully-connected network, and e) after the softmax operation in the scaled dot-product attention (see Sec. 6.1).

3.2 Event Proposal Generation Module

The proposal generator generates a set of proposals for a given video. It consists of two blocks: a bi-modal encoder and *bi-modal multi-headed proposal generator* (not related to multi-*headed* attention). The bi-modal encoder in this module inputs the whole sequence opposed to the bi-modal encoder in the captioning module, which inputs a sequence of features corresponding to a proposal. Specifically, it inputs both: visual-attended audio features $A^v \in \mathbb{R}^{T_a \times d_a}$ and audio-attended visual features $V^a \in \mathbb{R}^{T_v \times d_v}$. Since the sequence lengths (T_a, T_v) might be distinct, the fusion of predictions cannot be done at each time-step. To this end, we propose the module which makes predictions for each modality at every timestamp individually forming a common pool of cross-modal predictions (see Fig. 3).

Proposal Generation Head The proposal generation head inputs a sequence of T features, and makes predictions at each timestamp on the interval $[1, T]$, and for every prior segment length *anchor* in the set Ψ. The design of the proposal generation head is partly inspired by *YOLO* object detector [34, 35, 36]. Specifically, it is a *fully-convolutional* network which, in our case, consists of only three layers. Opposed to YOLO, we preserve the sequence length across all layers using *padding* and identity *stride*. Moreover, YOLO utilizes predictions from three different scales to predict different-scale objects. Hence, only three sizes of receptive fields are used. Instead, our model makes predictions at a single scale while controlling the receptive field with a *kernel size* k which is distinct in each proposal generation head. More precisely, the 1st convolutional layer has a kernel size k while in the 2nd and the 3rd the kernel size is 1. The layers are separated with *ReLU* activations and dropout.

Predictions Temporal boundaries and confidence for a proposal are obtained using three values which were predicted by the proposal generation head: a location of a segment center

$\sigma(c)$ relative to a position p in the sequence while $\sigma(\cdot)$ is a sigmoid function which bounds the values into [0, 1] interval, a coefficient $\exp(l)$ for an anchor, and *objectness score* $\sigma(o)$

$$\text{center} = p + \sigma(c); \qquad \text{length} = \text{anchor} \cdot \exp(l); \qquad \text{confidence} = \sigma(o). \tag{12}$$

The prediction of the center and length are in grid-cells (not in seconds). To obtain seconds, both are multiplied by a cell size which corresponds to a temporal span of the feature.

Bi-modal Multi-headed Proposal Generator The common pool of predictions is formed with predictions made by each of the proposal generation heads. Specifically, our model has K_a and K_v heads for audio and visual modalities with distinct sets of kernel sizes. Overall, our model generates $\left(T_a \cdot K_a \cdot |\Psi_a| + T_v \cdot K_v \cdot |\Psi_v|\right)$ proposals. For the final predictions, we select top-100 proposals out of the common pool based on the confidence score.

Segment Length Priors & Kernel Sizes To select a set of anchors, we use *K-Means* clustering algorithm with the *Euclidean distance* metric, as opposed to *intersection over the union* in YOLO. Due to granularity of feature extractors, feature lengths (T_a, T_v) might not necessarily equal. Thus, we obtain distinct numbers of anchors for audio and visual modalities $\left(|\Psi_a|, |\Psi_v|\right)$ to keep $T_a \cdot |\Psi_a|$ close to $T_v \cdot |\Psi_v|$ to balance the impact of each modality to the common pool of predictions. Similarly, the kernel sizes are determined by K-Means. We motivate it with an expectation that the receptive field will correspond to an event with a higher probability. We scale the resulting cluster centroids (in secs) by the feature time span to obtain values in grid-cell coordinates. Next, we round the values to the next odd integer for more elegant padding. Again, to preserve the balance in the share of predictions from each modality, we obtain an equal number of kernel sizes $K_a = K_v$ both modalities.

3.3 Training Procedure

Our model is trained in two stages: first, the captioning module is trained with ground truth proposals and, then, the proposal generator is trained using the pre-trained bi-modal encoder from the captioning model. Similar to [45] and [17], we optimize *KL-divergence* loss and apply *Label Smoothing* [43] to force a model to be less confident about predictions anticipating noisy annotations. Also, *masking* is used to ignore padding and prevent the model from attending to the next positions in the ground truth caption. During training of the event proposal generation module, all proposal generation heads for each modality are trained simultaneously summing up losses from all heads and both modalities. Each head uses YOLO-like loss: MSE for the localization losses (no square root) and *cross-entropy* for (no)objectness losses. The NMS is avoided for efficiency and to preserve the possibility of *dense* events. For the implementation details, a reader is referred to supplementary material (Sec. 6.3).

4 Experiments

We employ ActivityNet Captions dataset [20], which consists of 100k temporally localized sentences for 20k YouTube videos. The dataset is split into 50/25/25 % parts for training, validation, and testing. The validation set of videos is annotated by two different annotators. We report the results on the validation subsets as ground truth is not available for the testing set. Since the dataset is distributed as a set of links to YouTube videos, it is not possible to collect the whole dataset as some videos became unavailable. The authors also provide C3D features which are not suitable for our experimentation as they are missing audio information. In total, we had 91 % of the videos. We omit the unavailable videos from the validation

	RL	Full Dataset was Available	GT Proposals B@3	GT Proposals B@4	GT Proposals M	Learned Proposals B@3	Learned Proposals B@4	Learned Proposals M
Li *et al.* [22]	yes	yes	4.55	1.62	10.33	2.27	0.73	6.93
Xiong *et al.* [53]	yes	yes	–	–	–	2.84	1.24	7.08
Mun *et al.* [26]	yes	yes	4.41	1.28	**13.07**	**2.94**	0.93	**8.82**
Krishna *et al.* [20]	no	yes	4.09	1.60	8.88	1.90	0.71	5.69
Li *et al.* [22]	no	yes	4.51	1.71	9.31	2.05	0.74	6.14
Zhou *et al.* [62]	no	yes	**5.76**	**2.71**	**11.16**	2.91	**1.44**	6.91
Wang *et al.* [49]	no	yes	–	–	10.89	2.27	1.13	6.10
Mun *et al.* [26]	no	yes	–	–	–	–	–	6.92
Iashin *et al.* [17]	no	no	4.52	1.98	11.07	2.53	1.01	7.46
Rahman *et al.* [33]	no	no	3.04	1.46	7.23	1.85	0.90	4.93
Ours	no	no	**4.63**	**1.99**	10.90	**3.84**	**1.88**	**8.44**

Table 1: Comparison with state-of-the-art results on the dense video captioning task. The results are reported on the validation subset of ActivityNet Captions in both settings: captioning ground truth (GT) and learned proposals on BLEU@3–4 (B@3–4) and METEOR (M) metrics. For a fair comparison on METEOR, we additionally report the results of models without the reward (METEOR) maximization (RL) and indicate whether full dataset was available for training. The best and the 2$^{\text{nd}}$ best results are highlighted.

sets. We compared the results of other methods on the 91 % and 100 % of videos in Sec. 6.4.1 and observed similar performance suggesting the videos to be *missing completely at random*.

To evaluate the event proposal generation module we employ precision, recall, and mainly rely on F1-score (harmonic mean of precision and recall). While METEOR [7] and BLEU@3–4 [29] were used for captioning as they are highly correlated with human judgement. All metrics are averaged for every video and *temporal Intersection over Union* thresholds: $[0.3, 0.5, 0.7, 0.9]$. As it has been noted in [26], the original evaluation script had a critical issue which resulted in an incorrect evaluation of previous models. Therefore, we re-implement [49, 62] and compare with the results obtained with the corrected script.

4.1 Comparison to the State-of-the-art

We present the comparison between the bi-modal transformer with multi-headed proposal generator (Ours) and other methods in the existing literature [17, 20, 22, 26, 33, 49, 53, 62] on the dense video captioning task. The results of the comparison for captioning both ground truth (GT) and learned proposals are shown in Tab. 1. Since evaluating captioning is still challenging and METEOR is probably the best among other options, yet it only provides a *proxy* for how good a caption is. Therefore we believe that the direct optimization of METEOR using a reinforcement learning technique (RL) might not necessarily result in a better caption. To this end, we also include the results of [22, 26] without the RL module. Moreover, we obtained the results of [17] on the same subset of videos as we have since they additionally removed the videos with no speech modality from the evaluation.

According to the results, in the learned proposals setup, our dense video captioning model outperforms all of the models, which have no reward maximization on METEOR (no RL) while being on par when captioning ground truth proposals. Notably, our model has

	Full Dataset was Available	Prec.	Rec.	F1
Xiong *et al.* [53]	yes	**51.41**	24.31	33.01
Wang *et al.* [49]	yes	44.80	57.60	50.40
Zhou *et al.* [62]	yes	38.57	**86.33**	53.31
Mun *et al.* [26]	yes	**57.57**	55.58	**56.56**
Ours	no	48.23	**80.31**	**60.27**

Table 2: Comparison with state-of-the-art proposal generation methods on dense video captioning task. Results are reported on the validation set of ActivityNet Captions. Metrics: Precision, Recall, & F1-measure. The top-2 is highlighted.

the highest BLEU metrics in the learned proposal setup yet lies far away from [62] when captioning ground truth proposals on BLEU and performs on par with this model on METEOR.

Comparing to the RL methods, our model still outperforms them on BLEU metrics in both setups but loses in METEOR due to the absence of reward-maximization module. We draw the attention of a reader to the performance of [22] with and without the RL module — METEOR has dropped significantly yet other metrics remained on the same level.

Interestingly, we also outperform [17] who also use the transformer in multi-modal setup yet has more parameters (149M vs 51M). We note again that the results are not fair to neither of [17, 33] and ours since models have been trained on fewer videos.

Next, we compare our bi-modal multi-headed proposal generation module with other proposal generation modules from other dense video captioning models. The results for [62] and [49] are reported for 100 proposals per video. The results of the comparison are presented in Tab. 2. Despite our model being trained on fewer videos, our proposal generation model achieves state-of-the-art performance on the F1 metric. Specifically, our model provides impressive ground truth segment coverage while being accurate in its predictions.

4.2 Ablation Study

In this section, we show how the training procedure and modality impact the final results. The results are presented in Tab. 3 for both settings: captioning ground truth (performance of the captioning module) and leaned proposal (full dense video captioning model).

Training Procedures Our final model is trained in the following way. First, we train the captioning model on the ground truth proposal. Second, we freeze the weights on the encoder and train the proposal generator using the frozen encoder. The final results are obtained by captioning the proposals obtained from the trained proposal generator. Hence, the acronym "Cap $\rightarrow$ Prop" which reads as: "the proposal generator is trained using the pre-trained encoder from the captioning module". We compare this training procedure to other two methods: a) when both captioning and proposal generator modules are trained separately and b) when, first, the proposal module is trained and, then, the captioning module uses the pre-trained encoder with frozen weights during training. This is the opposite of the training procedure used for the final model, thus, abbreviated to "Prop $\rightarrow$ Cap".

Different Sets of Modalities The final model uses both audio and visual modalities to make predictions. We compare the performance of a bi-modal model with uni-modal ones. Specifically, for uni-modal settings, we employ the uni-modal transformer architecture similar to one in [17]. The difference between the hyper-parameters used for the final model and the uni-modal transformer is in the input dimension. For the uni-modal transformer, we follow the original paper where the input is first embedded into D_q dimension (see (14)) and

Training Procedure	Modality	GT Proposals B@3	GT Proposals B@4	GT Proposals M	Learned proposals B@3	Learned proposals B@4	Learned proposals M
Separately	Audio	2.85	1.14	8.81	2.50	1.11	6.89
	Visual	3.77	1.66	10.29	2.94	1.36	7.69
	Bi-modal	4.62	1.99	10.89	3.47	1.65	8.05
Prop $\rightarrow$ Cap	Audio	2.59	0.99	8.81	2.23	0.93	6.88
	Visual	3.62	1.56	10.16	3.08	1.45	7.81
	Bi-modal	4.10	1.78	10.48	3.07	1.47	7.67
Cap $\rightarrow$ Prop	Audio	2.85	1.14	8.81	2.58	1.15	6.98
	Visual	3.77	1.66	10.29	2.85	1.30	7.47
	Bi-modal	4.62	1.99	10.89	3.84	1.88	8.44

Table 3: The impact of training procedures and input modalities. We compare the training procedure of the final model when the proposal generator uses the pre-trained encoder on the captioning task (“Cap $\rightarrow$ Prop”) to an opposite scenario (“Prop $\rightarrow$ Cap”), and the situation when both of them are trained separately. The results are shown on validation sets of ActivityNet Captions when captioning ground truth (GT) and learned proposals.

remains the same everywhere later. We select 1024 for visual-only and 128 for audio-only transformers; the size of the pre-trained GloVe is projected with a FC layer to match the size.

Results We report every combination of the settings in Tab. 3. Specifically, we observed that the captioning module does not benefit from the pre-training for the proposal generation (“Prop $\rightarrow$ Cap” vs “Cap $\rightarrow$ Prop” & “Separate”). The results of the learned proposal setting show the importance of the pre-training but only in the “Cap $\rightarrow$ Prop” setting. Overall, we claim that the captioning training does not benefit from utilizing the pre-trained proposal generator’s encoder and, even, performs worse with it. While, the proposal generator ends up with better performance if pre-trained captioning module’s encoder is used.

The comparison of the cross-modal performance shows that using both modalities (audio and visual) gives the best result in nearly all cases in both settings. However, it is shown that the audio modality is the *weakest* among the three implying that visual modality might contain a stronger signal for video understanding. Nevertheless, the gap between the visual-only and bi-modal case is consistent in all settings. This suggests that the audio still provides essential cues for dense video captioning. More ablations studies can be found in Sec. 6.4.

5 Conclusion

We believe that the handling of multiple modalities is under-explored in the computer vision community. In this paper, we present a novel bi-modal transformer with a bi-modal multi-headed proposal generation module showing how audio might facilitate the performance of dense video captioning. We perform our experimentation on the ActivityNet Captions dataset and achieve state-of-the-art results on F1 and BLEU metrics. The the ablation study results show that the proposed model provides an effective and elegant way of fusing audio and visual features while outperforming the uni-modal configurations in all settings.

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

References

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

[2] Hedi Ben-Younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. MUTAN: Multimodal tucker fusion for visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2612–2620, 2017.

[3] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. SST: Single-stream temporal action proposals. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2911–2920, 2017.

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[5] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[6] Pradipto Das, Chenliang Xu, Richard F Doell, and Jason J Corso. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2634–2641, 2013.

[7] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380, 2014.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423.

[9] Xuguang Duan, Wenbing Huang, Chuang Gan, Jingdong Wang, Wenwu Zhu, and Junzhou Huang. Weakly supervised dense event captioning in videos. In *Advances in Neural Information Processing Systems*, pages 3059–3069, 2018.

[10] Victor Escorcia, Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. DAPs: Deep action proposals for action understanding. In *European Conference on Computer Vision*, pages 768–784, 2016.

[11] Zhe Gan, Chuang Gan, Xiaodong He, Yunchen Pu, Kenneth Tran, Jianfeng Gao, Lawrence Carin, and Li Deng. Semantic compositional networks for visual captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5630–5639, 2017.

[12] Jort F Gemmeke, Daniel PW 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 *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780, 2017.

[13] Wangli Hao, Zhaoxiang Zhang, and He Guan. Integrating both visual and audio cues for enhanced video caption. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135, 2017.

[16] Chiori Hori, Takaaki Hori, Teng-Yok Lee, Ziming Zhang, Bret Harsham, John R Hershey, Tim K Marks, and Kazuhiko Sumi. Attention-based multimodal fusion for video description. In *Proceedings of the IEEE international conference on computer vision*, pages 4193–4202, 2017.

[17] Vladimir Iashin and Esa Rahtu. Multi-modal dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 958–959, 2020.

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

[19] Atsuhiro Kojima, Takeshi Tamura, and Kunio Fukunaga. Natural language description of human activities from video images based on concept hierarchy of actions. *International Journal of Computer Vision*, 50(2):171–184, 2002.

[20] 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*, pages 706–715, 2017.

[21] Mun Wai Lee, Asaad Hakeem, Niels Haering, and Song-Chun Zhu. Save: A framework for semantic annotation of visual events. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8, 2008.

[22] 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*, pages 7492–7500, 2018.

[23] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *European conference on computer vision*, pages 21–37, 2016.

[24] Xiang Long, Chuang Gan, Gerard De Melo, Jiajun Wu, Xiao Liu, and Shilei Wen. Attention clusters: Purely attention based local feature integration for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7834–7843, 2018.

[25] 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 *Proceedings of the IEEE international conference on computer vision*, pages 2630–2640, 2019.

[26] Jonghwan Mun, Linjie Yang, Zhou Ren, Ning Xu, and Bohyung Han. Streamlined dense video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6588–6597, 2019.

[27] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 631–648, 2018.

[28] Yingwei Pan, Ting Yao, Houqiang Li, and Tao Mei. Video captioning with transferred semantic attributes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6504–6512, 2017.

[29] 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 of the Association for Computational Linguistics*, pages 311–318, 2002.

[30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.

[31] Wenjie Pei, Jiyuan Zhang, Xiangrong Wang, Lei Ke, Xiaoyong Shen, and Yu-Wing Tai. Memory-attended recurrent network for video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8347–8356, 2019.

[32] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[33] Tanzila Rahman, Bicheng Xu, and Leonid Sigal. Watch, listen and tell: Multi-modal weakly supervised dense event captioning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8908–8917, 2019.

[34] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[35] Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. *arXiv*, 2018.

[36] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[37] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024, 2017.

[38] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description. *International Journal of Computer Vision*, 123(1):94–120, 2017.

[39] Marcus Rohrbach, Wei Qiu, Ivan Titov, Stefan Thater, Manfred Pinkal, and Bernt Schiele. Translating video content to natural language descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 433–440, 2013.

[40] Botian Shi, Lei Ji, Yaobo Liang, Nan Duan, Peng Chen, Zhendong Niu, and Ming Zhou. Dense procedure captioning in narrated instructional videos. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6382–6391, 2019.

[41] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[42] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018.

[43] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[44] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[46] 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*, pages 4534–4542, 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 *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1494–1504, Denver, Colorado, May–June 2015. Association for Computational Linguistics. doi: 10.3115/v1/N15-1173.

[48] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in neural information processing systems*, pages 2692–2700, 2015.

[49] Jingwen Wang, Wenhao Jiang, Lin Ma, Wei Liu, and Yong Xu. Bidirectional attentive fusion with context gating for dense video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7190–7198, 2018.

[50] Junbo Wang, Wei Wang, Yan Huang, Liang Wang, and Tieniu Tan. M3: Multimodal memory modelling for video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7512–7520, 2018.

[51] Xin Wang, Wenhu Chen, Jiawei Wu, Yuan-Fang Wang, and William Yang Wang. Video captioning via hierarchical reinforcement learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4213–4222, 2018.

[52] Xin Wang, Yuan-Fang Wang, and William Yang Wang. Watch, listen, and describe: Globally and locally aligned cross-modal attentions for video captioning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 795–801, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2125.

[53] Yilei Xiong, Bo Dai, and Dahua Lin. Move forward and tell: A progressive generator of video descriptions. In *Proceedings of the European Conference on Computer Vision*, pages 468–483, 2018.

[54] Jun Xu, Ting Yao, Yongdong Zhang, and Tao Mei. Learning multimodal attention lstm networks for video captioning. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 537–545, 2017.

[55] Chenggang Yan, Yunbin Tu, Xingzheng Wang, Yongbing Zhang, Xinhong Hao, Yongdong Zhang, and Qionghai Dai. STAT: spatial-temporal attention mechanism for video captioning. *IEEE Transactions on Multimedia*, 22(1):229–241, 2019.

[56] 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*, pages 4507–4515, 2015.

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

[58] 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*, pages 4584–4593, 2016.

[59] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels. In *Proceedings of the European conference on computer vision (ECCV)*, pages 570–586, 2018.

[60] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2914–2923, 2017.

[61] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[62] 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*, pages 8739–8748, 2018.

[63] Lingyu Zhu and Esa Rahtu. Separating sounds from a single image. *arXiv preprint arXiv:2007.07984*, 2020.

[64] Lingyu Zhu and Esa Rahtu. Visually guided sound source separation using cascaded opponent filter network. *arXiv preprint arXiv:2006.03028*, 2020.