Bidirectional Attentive Fusion with Context Gating for Dense Video Captioning

Jingwen Wang[†*] Wenhao Jiang[‡§] Lin Ma[‡§] Wei Liu[‡] Yong Xu[†]
[†]South China University of Technology [‡]Tencent AI Lab
{jaywongjaywong, cswhjiang, forest.linma}@gmail.com
wliu@ee.columbia.edu yxu@scut.edu.cn

Abstract

*Dense video captioning is a newly emerging task that aims at both localizing and describing all events in a video. We identify and tackle two challenges on this task, namely, (1) how to utilize both past and future contexts for accurate event proposal predictions, and (2) how to construct informative input to the decoder for generating natural event descriptions. First, previous works predominantly generate temporal event proposals in the forward direction, which neglects future video context. We propose a bidirectional proposal method that effectively exploits both past and future contexts to make proposal predictions. Second, different events ending at (nearly) the same time are indistinguishable in the previous works, resulting in the same captions. We solve this problem by representing each event with an attentive fusion of hidden states from the proposal module and video contents (e.g., C3D features). We further propose a novel context gating mechanism to balance the contributions from the current event and its surrounding contexts dynamically. We empirically show that our attentively fused event representation is superior to the proposal hidden states or video contents alone. By coupling proposal and captioning modules into one unified framework, our model outperforms the state-of-the-arts on the ActivityNet Captions dataset with a relative gain of over **100%** (Meteor score increases from **4.82** to **9.65**).*

1. Introduction

With the rapid growing of videos on the Internet, it becomes much more important to automatically classify and retrieve these videos. While images and short videos have attracted extensive attentions from vision research community [43, 25, 40, 30, 15, 9, 31, 6, 24], understanding long untrimmed videos remains an open question. To help further understand videos and bridge them with human language, a new task of dense video captioning is proposed [20]. The goal is to automatically localize events in videos and describe each one with a sentence. The capability of localizing and describing events in videos will benefit a broad range of applications, such as video summarization [23, 29], video retrieval [35, 44], video object detection [48], video segment localization with language query [1, 11], and so on.

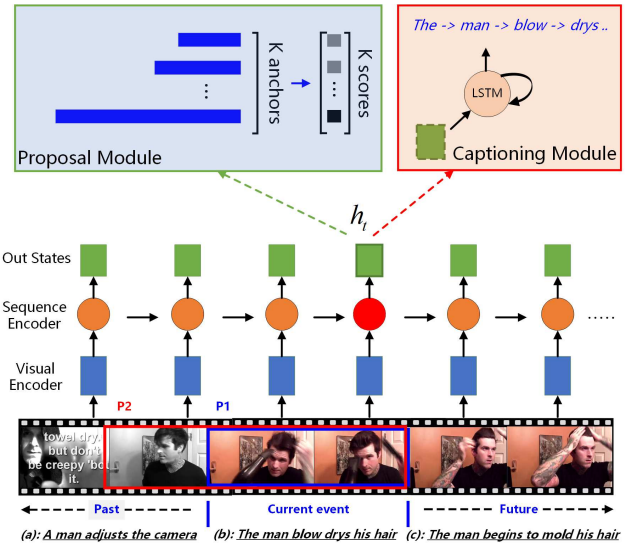


Figure 1. Two main challenges in dense video captioning. First, previous works, *e.g.*, SST [3], process a video sequence in the forward direction. Future video context (c) expressing "`The man begins to mold his hair`" is not considered, which presents close relationship with current proposal (b) expressing "`The man blow drys his hair`". Second, previous work only uses the proposal hidden state h_t at time step t to represent the detected proposal, which cannot distinguish events (*e.g.*, P1, P2) that end at the same time step.

Compared to video captioning, which targets at describing a short video clip (e.g., 20s long in MSR-VTT dataset [41]), dense video captioning requires analyzing a much longer and complicated video sequence (e.g., 120s long in ActivityNet Captions [20]). Since long videos usually involve multiple events, dense video captioning requires simultaneously performing temporal localization and

*Work done while Jingwen Wang was a Research Intern with Tencent AI Lab.
§Corresponding authors.

captioning, which issues the following two challenges.

First, generating video action proposals requires localizing all possible events that occur in a video. To do so, one simple way would be to use sliding windows to iterate over a video and classify every window to either an action or background. However, this kind of methods can only produce short proposals that are no longer than the predefined sliding window. To overcome this problem, Buch *et al.* [3] proposes Single Stream Temporal Action Proposals (SST) to eliminate the need to divide long video sequences into clips or overlapped temporal windows. As shown in Fig. 1, SST runs through a video only once and densely makes proposal predictions ending at that time step, with k different offsets. Krishna *et al.* [20] uses a similar proposal method as SST. While promising results were achieved, these methods simply ignore future event context and only encode past context and current event information to make predictions. Since events happening in a video are usually highly correlated, it is non-preferable to discard valuable future information. For example, in Fig. 1, when making proposal prediction at the end of event (b), SST has run over both past context (a) and current event content, but not future video context (c). Event (b) highly correlates with event (c). Recognizing and localizing event (b) will help localize event (c) more accurately, and vice versa. In this paper, we propose a straightforward yet effective solution, namely, Bidirectional SST, towards efficiently encoding both past, current, and future video information. Specifically, in the forward pass we learn k binary classifiers corresponding to k anchors densely at each time step; in backward pass we reverse both video sequence input and predict proposals backwards. This means that the forward pass encodes past context and current event information, while the backward pass encodes future context and current event information. Finally we merge proposal scores for the same predictions from the two passes and output final proposals. Technical details can be found in Section 3.

Once proposals are obtained, another important question is how to represent these proposals in order to generate language descriptions. In [20], the LSTM hidden state in proposal module is reused to represent a detected proposal. However, the discrimination property of event representation is overlooked. As shown in Fig. 1, k proposals (anchors) end at same time step, but only one LSTM hidden state h_t at that time step is returned. For example, $P1$ and $P2$ will be both represented with h_t. To construct more discriminative proposal representation, we propose to fuse proposal state information and detected video content (e.g. C3D sequences). The intuition behind that is involving detected clips help discriminate highly overlapped events, since the detected temporal regions are different. Based on this idea, we further explore several ways for fusing these two kinds of information to boost dense captioning performance.

To output more confident results, we further propose joint ranking technique to select high-confidence proposal-caption pairs by taking both proposal score and caption confidence into consideration.

To summarize, the contributions of this paper are three-fold. First, we present Bidirectional SST for better temporal action proposals with both past, current, and future contexts encoded. Second, for captioning module, we explore different ways to attentively fuse proposal state information and detected video content to effectively discriminate highly overlapped events. Third, we further present joint ranking at inference time to select proposal-caption pairs with high confidence.

2. Related Work

Dense video captioning requires both temporally localization and descriptions for all events happening in a video. These two tasks can be handled as pipelines or coupled together for end-to-end processing. We review related works on the above two tasks.

2.1. Temporal Action Proposals

Analogous to region proposals in image domain, temporal action proposals are candidate temporal windows that possibly contain actions. Sparse-prop [4] applies dictionary learning for generating class-independent proposals. S-CNN [34] uses 3D convolutional neural networks (CNNs) [36] to generate multi-scale segments (proposals). TURN TAP [12] uses clip pyramid features in theri model, and it predicts proposals and refines temporal boundaries jointly. DAPs [8] first applies Long Short-Term Memory (LSTM) [14] to encoding video content in a sliding window and then predicts proposals covered by the window. Built on [8], SST [3] further takes long sequence training problem into consideration and generates proposals in a single pass. However, all these methods either fail to produce long proposals or do not exploit future context. In contrast, our model for temporal proposal tackles these two problems simultaneously.

2.2. Video Captioning

Video captioning with one single sentence. There are a large body of works on this topic. Earlier works are template-based [13, 33], which replace POS (part-of-speech) tags with detected objects, attributes, and places. [13] learns semantic hierarchies from video data in order to choose an appropriate level of sentence descriptions. [33] first formulates video captioning as a machine translation problem and uses CRF to model semantic relationship between visual components. Recent approaches are neural-based, in an encoder-decoder fashion [19, 38, 26, 45, 49, 27, 10, 7, 22]. Venugopalan *et al.* models both video

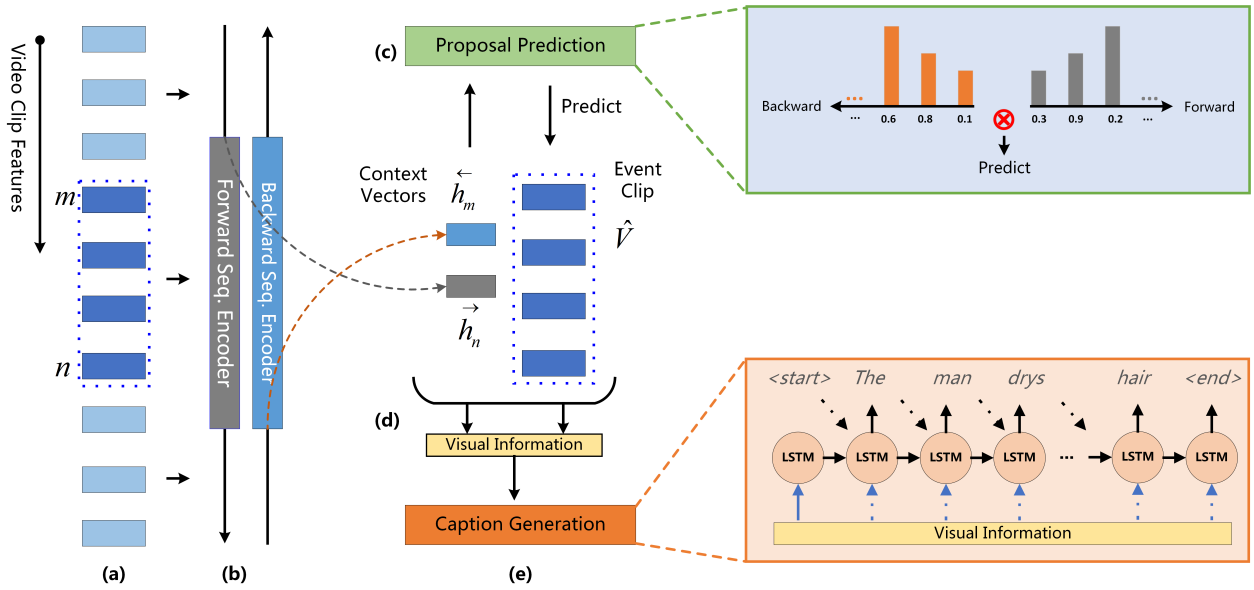


Figure 2. The main framework of our proposed method. (a) A video input is first encoded as a sequence of visual features (e.g., C3D). (b) The visual features are then fed into our bidirectional sequence encoder (e.g., LSTM). (c) Each hidden state from the forward/backward seq. encoder will be fed into the proposal module. The forward/backward seq. encoders are jointly learned to make proposal predictions. (d) Hidden states at boundary of a detected event ($\overrightarrow{\mathbf{h}}_n, \overleftarrow{\mathbf{h}}_m$) will be served as context vectors for the event. The context vectors and detected event clip features are then fused together and served as visual information input. We detail the fusion methods in Section 3.2.2. (e) The decoder LSTM translates visual input into a sentence.

and language as sequences using recurrent neural networks [38]. To strengthen the semantic relationships between a video and corresponding generated sentence, Pan *et al.* proposed to learn a translation and a common embedding space shared by video and language jointly [26]. Some subsequent methods further explore attention models in video context. Inspired by the soft attention mechanism [42] in image captioning, Yao *et al.* proposed to generate temporal attention over video frames when predicting next word [45]. Zhang *et al.* proposed to learn a task-driven fusion model by dynamically fusing complementary features from multiple channels (appearance, motion) [49]. Some other works [27, 10, 46] exploit attributes or concepts (objects, actions, etc.) to improve video captioning performance. Chen *et al.* further considered different topics from web videos and generating topic-guided descriptions [7].

Video captioning with a paragraph. While aforementioned captioning methods generate only one sentence for the input video, video paragraph generation focuses on producing multiple semantics-fluent sentences. Rohrbach *et al.* adapted statistical machine translation (SMT) [33] to generate semantic consistent sentences with desired level of details [32]. Yu *et al.* proposed a hierarchical RNN to model both cross-sentence dependency and word dependency [47].

Dense video captioning. Video paragraph generation relies on alignment from *ground-truth event intervals* at test time. To relieve this constraint, dense video captioning generates multiple sentences and grounds them with time locations automatically, which is thus much more challenging. To the best of our knowledge, [20] is the only published work on this topic. In [20], the task of dense-captioning events in video together with a new dataset: ActivityNet Captions[1] were introduced. The model in [20] is composed of an event proposal module and a captioning module. The event proposal module detects events with a multi-scale version of DAPs [8] and represents them with LSTM hidden states. The captioning module is responsible for describing each detected proposal. Compared to [20], our method enjoys the following advantages. First, our bidirectional proposal module encodes both past and future contexts while [20] only utilizes past context for proposal prediction. Second, our model is able to distinguish and describe highly overlapped events while [20] cannot.

3. Method

In this section we introduce our main framework for densely describing events in videos, as shown in Fig. 2. We will first introduce our bidirectional proposal module, then our captioning module. Note that these two modules couple together and thus can be trained in an end-to-end manner.

[1] http://cs.stanford.edu/people/ranjaykrishna/densevid/

3.1. Proposal Module

The goal of the proposal module is to generate a set of temporal regions that possibly contain actions or events. Formally, assume that we have a video sequence $X = \{x_1, x_2, ..., x_L\}$ with L frames. Following [20], each video frame is encoded by the 3D CNN [36], which was pretrained on Sports-1M video dataset [17]. The extracted C3D features are of temporal resolution δ = 16 frames, discretizing the input stream into $T = L/\delta$ time steps. We perform PCA to reduce the feature dimensionality (from 4096 to 500). The generated visual stream is thus $V = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_T\}$.

Forward Pass. We use LSTM to sequentially encode the visual stream. The sequence encoder processes visual sequences and accumulates visual clues across time. The output LSTM hidden state $\overrightarrow{\mathbf{h}}_t \in \{\overrightarrow{\mathbf{h}}_i\}_{i=1}^{T}$ at time step t thus encodes visual information for the passed time steps $\{1, 2, \ldots, t\}$. The hidden state will be fed into K independent binary classifiers and produces K confidence scores $\overrightarrow{C}_p^{\,t} = \{\overrightarrow{c}_i^{\,t}\}_{i=1,...,K}$ indicating the probability of K proposals specified by $\overrightarrow{S}^t = \{\overrightarrow{s}_i^{\,t}\}_{i=1,...,K}$. $\overrightarrow{s}_i^{\,t}$ denotes a video clip with end time as t and start time as $t - l_i$, where $\{l_i\}_{i=1}^{K}$ is the lengths of the predefined K proposal anchors. Please note that all the K proposals in $\overrightarrow{S}^t$ share the same end time t. The proposal scores $\overrightarrow{C}_p^{\,t}$ is calculated by a fully connected layer:

$$\overrightarrow{C}_p^{\,t} = \sigma(\overrightarrow{W}_c \overrightarrow{\mathbf{h}}_t + \mathbf{b}_c), \tag{1}$$

where σ denotes *sigmoid* nonlinearity. $\overrightarrow{W}_c$, $\mathbf{b}_c$ are shared across all time steps.

Backward Pass. Our proposed bidirectional proposal module also involves a backward pass. The aim of such a procedure is to capture future context, in addition to current event clue for better event proposals. We feed the input sequence V in a reverse order to the backward sequence encoder. It is expected to predict proposals with high scores at the original start time of proposals. Similarly, at each time step, we obtain K proposals $\overleftarrow{S}^t = \{\overleftarrow{s}_i^{\,t}\}_{i=1,...,K}$ with K confidence scores $\overleftarrow{C}_p^{\,t} = \{\overleftarrow{c}_i^{\,t}\}_{i=1,...,K}$, and a hidden state $\overleftarrow{\mathbf{h}}_t$.

Fusion. After the two passes, we obtain N proposals collected from all time steps of both directions. In order to select proposals with high confidence, we fuse the two sets of scores for the same proposals, yielding the final scores:

$$C_p = \{\overrightarrow{c}_i \times \overleftarrow{c}_i\}_{i=1}^{N}. \tag{2}$$

Many fusing strategies can be adopted. In this paper, we simply use the multiplication to fuse proposals from the two passes together. Proposals with scores larger than a threshold τ will be finally selected for further captioning. We do not perform non-maximum suppression since events happening in a video are usually highly overlapped, the same as what has been adopted in [20].

3.2. Captioning Module

Following the encoder-decoder framework, a recurrent neural network, specifically LSTM, is leveraged in our captioning module to translate visual input into a sentence. In this section, we first recap LSTM. Then we describe a novel dynamic fusion method.

3.2.1 Decoder: Long Short-Term Memory

LSTM [14] is used as our basic building block, considering its excellent ability for modeling sequences. An LSTM unit consists of an input cell g_t, an input gate i_t, a forget gate f_t, and an output gate o_t and they can be computed by:

$$\begin{pmatrix} i_t \\ f_t \\ o_t \\ g_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} \mathbf{E}_t \\ \mathbf{F}_t \\ \mathbf{H}_{t-1} \end{pmatrix}, \tag{3}$$

where $\mathbf{E}_t$ is the embedding of input word at time step t, $\mathbf{F}_t$ is representation at t that will be described later, $\mathbf{H}_{t-1}$ is the previous LSTM hidden state and W is a transformation matrix to be learned. The memory cell c_t and hidden state $\mathbf{H}_t$ are updated by:

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t, \tag{4}$$

$$\mathbf{H}_t = o_t \odot \tanh(c_t), \tag{5}$$

where $\odot$ denotes element-wise multiplication operator. At each time step, a linear projection and softmax operation are performed on the hidden state to generate probability distribution over all possible words.

3.2.2 Dynamic Attentive Fusion with Context Gating

To caption a detected proposal, previous work just takes the proposal hidden state as input to the LSTM [20]. In this paper we propose to fuse the proposal states from the forward and backward passes, which capture both past and future contexts, together with the encoded visual features of the detected proposal. Formally, the visual input to the decoder is:

$$\mathbf{F}_t(s_i) = f(\overrightarrow{\mathbf{h}}_n, \overleftarrow{\mathbf{h}}_m, \hat{V} = \{\mathbf{v}_i\}_{i=m}^{n}, \mathbf{H}_{t-1}), \tag{6}$$

where m and n denote the start and end time stamp for the detected event s_i. $\hat{V}$ denotes the clip features, specifically

C3D for the proposal s_i. $\overrightarrow{\mathbf{h}}_n$ and $\overleftarrow{\mathbf{h}}_m$ are the proposal hidden states, encoding the past and future context information of the detected proposal, which are simply named context vectors. $\mathbf{H}_{t-1}$ is the previous LSTM hidden state. And f is a mapping to output a compact vector, which is to be fed into LSTM unit using Eq. (3).

The most straightforward way is to simply concatenate $\hat{V}$, $\overrightarrow{\mathbf{h}}_n$, and $\overleftarrow{\mathbf{h}}_m$ together without considering $\mathbf{H}_{t-1}$. However, it is implausible, as the dimension of $\hat{V}$ depends on the length of a detected event. Another simple way is to use the mean of $\hat{V}$ and concatenate it with proposal hidden states. However, mean pooling does not explicitly explore relationship between an event and surrounding contexts.

Temporal Dynamic Attention. As demonstrated in [42, 45], dynamically attending on image sub-regions and video frames at each time step when decoding can effectively improve captioning performance. Therefore, in our dense captioning model, we also design a dynamic attention mechanism to fuse visual features $\hat{V}$ and context vectors $\overrightarrow{\mathbf{h}}_n$, $\overleftarrow{\mathbf{h}}_m$. At each time step t, the relevance score z_i^t for $\mathbf{v}_{i+m-1}$ is obtained by:

$$z_i^t = W_a^T \cdot \tanh(W_{\mathbf{v}}\mathbf{v}_{i+m-1} + W_h[\overrightarrow{\mathbf{h}}_n, \overleftarrow{\mathbf{h}}_m] + W_{\mathbf{H}}\mathbf{H}_{t-1} + \mathbf{b}), \tag{7}$$

where $\mathbf{H}_{t-1}$ is the hidden states of decoder at the $t-1$ time step. $[\cdot,\cdot]$ denotes vector concatenation. The weights of $\mathbf{v}_{i+m-1}$ can be obtained by a softmax normalization:

$$\alpha_i^t = \exp(z_i^t) / \sum_{k=1}^{p} \exp(z_k^t), \tag{8}$$

where $p = n - m + 1$ denotes the length of a proposal. The attended visual feature is generated by a weighted sum:

$$\tilde{\mathbf{v}}^t = \sum_{i=1}^{p} \alpha_i^t \cdot \mathbf{v}_{i+m-1}. \tag{9}$$

We expect the model can better locate "key frames" and produce more semantic correlated words by involving context vectors for calculating the attention as in Eq. (7). The final input to LSTM unit could be expressed as:

$$\mathbf{F}(s_i) = [\tilde{\mathbf{v}}^t, \overrightarrow{\mathbf{h}}_n, \overleftarrow{\mathbf{h}}_m]. \tag{10}$$

Context Gating. Inspired by the gating mechanisms in LSTM, we propose to explicitly model the relative contributions of the attentive event feature and contexts when generating a word. Specifically, once obtain the attended visual feature $\tilde{\mathbf{v}}^t$, instead of directly concatenating it with context vectors, we learn a "context gate" function to balance them. In our context gating mechanism, the first step is to project the event feature and context vectors into the same space:

$$\dot{\mathbf{v}}^t = \tanh(\tilde{W}\tilde{\mathbf{v}}^t), \tag{11}$$

$$\mathbf{h} = \tanh(W_{ctx}[\overrightarrow{\mathbf{h}}_n, \overleftarrow{\mathbf{h}}_m]), \tag{12}$$

where $\tilde{W}$ and W_{ctx} are the projection matrices. The context gate is then calculated by a nonlinear layer:

$$g_{ctx} = \sigma(W_g[\dot{\mathbf{v}}^t, \mathbf{h}, \mathbf{E}_t, \mathbf{H}_{t-1}]), \tag{13}$$

where $\mathbf{E}_t$ is word embedding vector, $\mathbf{H}_{t-1}$ is previous LSTM state. The context gate explicitly measures the contribution for surrounding context information ($\mathbf{h}$) at current decoding stage (given $\mathbf{E}_t$, $\mathbf{H}_{t-1}$). We then use the context gate to fuse the event feature and the context vector together:

$$\mathbf{F}(s_i) = [(1 - g_{ctx}) \odot \dot{\mathbf{v}}^t, g_{ctx} \odot \mathbf{h}]. \tag{14}$$

With this mechanism, we expect the network to learn how much context should be used when generating next word.

3.3. Training

Our complete dense video captioning model, as illustrated in Fig. 2, couples the proposal and captioning module together. Therefore, two types of loss functions are considered in our model, specifically, the proposal loss and captioning loss.

Proposal Loss. We collect lengths of all ground-truth proposals and group them into K=128 clusters (anchors). Each training example $V = \{\mathbf{v}_i\}_{i=1}^T$ is associated with ground-truth labels $\{y_t\}_{t=1}^T$. Each y_t is a K-dim vectors with binary entries. y_t^j is set to 1 if the corresponding proposal interval has a temporal Intersection-over-Union (tIoU) with the ground-truth larger than 0.5 and set to 0 otherwise. We adopt weighted multi-label cross entropy as proposal loss $\mathcal{L}_p$ following [3] to balance positive and negative proposals. For a given video $X \in \mathcal{X}$ at time step t:

$$\mathcal{L}_p(c, t, X, y) = -\sum_{j=1}^{K} w_0^j y_t^j log c_t^j + w_1^j (1 - y_t^j) log(1 - c_t^j), \tag{15}$$

where w_0^j, w_1^j are determined based on the numbers of positive and negative proposal samples. c_t^j is the prediction score for the j-th proposal at time t. We calculate forward and backward loss in the same way. We add them together and jointly train the forward and backward proposal module. $\mathcal{L}_p$ is obtained by averaged along time steps and for all videos.

Captioning Loss. We only feed proposals of high tIoU ($>$ 0.8) with ground-truths to train captioning module. Following [39], we define captioning loss $\mathcal{L}_c$ as sum of negative log likelihood of *correct word* in a sentence with M words:

$$\mathcal{L}_c(P) = -\sum_{i=1}^{M} \log(p(w_i)), \tag{16}$$

where w_i is the i-th word in a ground truth sentence. $\mathcal{L}_c$ is obtained by averaging all $\mathcal{L}_c(P)$ for all proposals P.

Total Loss. By considering both proposal localization and captioning, the total loss is given by:

$$\mathcal{L} = \lambda \times \mathcal{L}_p + \mathcal{L}_c, \tag{17}$$

where λ balances the contributions between proposal localization and captioning, which is simply set to 0.5. A two-layer LSTM is used to encode a video stream and densely predict K proposals at each time step. Without loss of generality, we do not use multiple strides but with only stride of 64 frames (2 seconds). This gives a reasonable number of unfolding LSTM steps (60 on average) to learn temporal dependency. We do not perform stream sampling and only take the whole video as a single stream, to make sure all ground truth proposals are included. For fair comparison, we initialize our bidirectional sequence encoder with a single layer LSTM for each direction (the baseline method adopts a two-layer LSTM). We use a two-layer LSTM for decoding stage (caption generation). We first train the proposal module (about 5 epochs) to ensure a good initialization and then train the whole model in an end-to-end manner. The Adam [18] optimization algorithm with base learning rate of 0.001 and batch size of 1 is used. *Please note that the above hyper parameters are the same for our method and all baselines.*

3.4. Inference by Joint Ranking

As illustrated in Fig. 2, dense captioning involves the two aforementioned modules. As such, to affectively describe each event, two conditions need to be satisfied: (1) the localization yielded by proposal module is of high score; (2) the produced caption is of high confidence. To this end, we propose a novel joint ranking approach for dense captioning during the inference stage. We use Eq. (2) to measure the proposal score C_p. For a generated caption of a proposal consisting of M words $\{w_i\}_{i=1}^{M}$, we define its confidence by summing all log probabilities of predicted words:

$$c_c = \sum_{i=1}^{M} \log(p(w_i)). \tag{18}$$

Larger $p(w_i)$ indicates higher confidence score. Let $C_c = \{c_c^{(i)}\}_{i=1}^{N}$ denotes confidence scores of all sentences. We merge the two scores with a weighted sum strategy by simultaneously considering proposal localization and captioning:

$$C = \gamma \times C_p + C_c, \tag{19}$$

where γ is a trade-off parameter to control the contributions from localization and captioning. As C_p is of smaller scale, γ is empirically set as 10 in this paper. Based on the obtained C, Top K proposals together with their captions are selected for further evaluation.

4. Experiment

To detail our contributions, we conduct experiments on the two tasks: event localization and dense event captioning. The former evaluates how good the generated proposals are, and the latter measures the performance of our whole dense captioning system. We begin by describing the benchmark dataset: ActivityNet Captions [20].

Dataset. ActivityNet Captions [20] is built on ActivityNet v1.3 [5] which includes 20k YouTube untrimmed videos from real life. The videos are 120 seconds long on average. Most of the videos contain over 3 annotated events with corresponding start/end time and human-written sentences, which contain 13.5 words on average. The number of videos in train/validation/test split is 10024/4926/5044, respectively. Ground truth annotations from the test split are withheld for competition. Therefore, we first compare our model with baseline methods on validation set, then we report our final result returned from the test server.

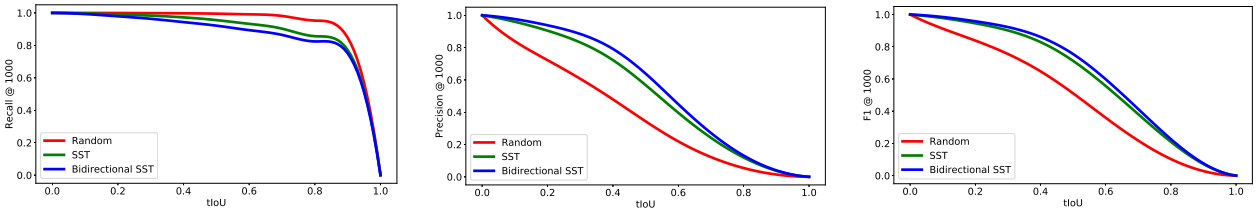
4.1. Event Localization

Metric. We use Precision@1000 and Recall@1000 averaged at different tIoU thresholds $\{0.3, 0.5, 0.7, 0.9\}$ as metrics. The evaluation toolkit we used is provided by [20]. We also use F1 score to simultaneously consider precision and recall, arguing that F1 is a more reasonable metric for event localization, by showing experimental evidences.

Compared Methods. We compare the following methods:

- Random: Both start time and end time are chosen randomly.
- Forward SST: The method used in [3].
- Backward SST: Similar as Forward SST, except that the video sequence is fed in a reverse order.
- Bidirectional SST: Our proposed method. We combine Forward SST and Backward SST and jointly inference by fusing scores for the same predicted proposals.

Results. As shown in Fig. 3, Random proposal method gives the highest recall rate among all compared methods. The reason is that most ground truth proposals are pretty long (30% compared to total video length on average, while only 2% for THUMOS-14 [16] action dataset), and thus randomly sampling can possibly cover all ground truth proposals. However, random sampling gives very low precision. A low-precision proposal method will cause performance degeneration for dense captioning system which simply describes all proposals. This is different from action detection, which involves a classification module to further filter out background proposals. Therefore, we mainly refer to F1 score which combines both precision and recall to measure how good the generated proposals are. We compare our bidirectional proposal module with baseline methods using F1 against different tIoU thresholds with



(a) Recall@1000 *vs.* tIoU (b) Precision@1000 *vs.* tIoU (c) F1@1000 *vs.* tIoU

Figure 3. Comparison for different proposal methods.

Table 1. Performance of different methods on ActivityNet Captions validation set. All values are reported in percentage (%). No validation result is provided by [20]. H: context vectors, E: event features, TDA: temporal dynamic attention, CG: context gate.

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	Meteor	Rouge-L	CIDEr-D
SST + H	16.78	5.94	2.22	0.88	7.87	16.75	8.17
Bi-SST + H	17.25	6.48	2.68	1.20	8.35	17.56	8.49
Bi-SST + E	17.51	7.17	3.08	1.32	8.36	17.96	9.13
Bi-SST + E + H	17.50	6.95	2.94	1.28	8.78	17.68	9.10
Bi-SST + E + H + TDA	18.70	8.17	3.63	1.59	9.00	18.64	10.02
Bi-SST + E + H + TDA + CG	**19.37**	8.69	4.03	1.89	9.19	**19.29**	11.03
Bi-SST + E + H + TDA + CG + Ranking	18.99	**8.84**	**4.41**	**2.30**	**9.60**	19.10	**12.68**

Table 2. Comparison with the state-of-art method on ActivityNet Captions test set. The test server returns only Meteor score (in percentage (%)).

Method	Meteor
Krishna *et al.* [20]	4.82
Ours	9.65

ground truth proposals. Our method surpasses SST with clear margins as shown in Fig. 3. This confirms that bidirectional prediction with encoded future context indeed improves proposal quality, compared to single direction prediction model.

4.2. Dense Event Captioning

Metric. We mainly refer to Meteor [2] to measure the similarity between two sentences as it is reported to be most correlated to human judgments when a small number of sentence references are given [37]. To measure the whole dense captioning system, we average Meteor scores at tIoU thresholds of 0.3, 0.5, 0.7, and 0.9 when describing top 1000 proposals for each video. The same strategy has been adopted in [20]. For validation split, we also provide BLEU [28], Rouge-L [21], and CIDEr-D [37] scores for complete comparison. For test split, we report Meteor score, since the test server only returns Meteor result.

Compared Methods. We denote "H" as context vectors, "E" as event clip features, "TDA" as temporal dynamic attention fusion, and "CG" as context gate, respectively. We compare the following methods:

- SST + H: This method utilizes SST [3] to generate proposals and represents them with corresponding hidden states for generating descriptions. This approach is served as our baseline.
- Bi-SST + H: We apply our bidirectional proposal method to generating proposals. The hidden states from both direction are concatenated to represent an event.
- Bi-SST + E: Mean pooled event feature is used to represent the detected event.
- Bi-SST + E + H: Mean pooled event feature and hidden states are concatenated for representation.
- Bi-SST + E + H + TDA: Temporal dynamic attention (TDA) is used to dynamically construct visual input to the decoder.
- Bi-SST + E + H + TDA + CG: Context gate is used to balance the attended event feature and contexts.
- Bi-SST + E + H + TDA + CG + Ranking: Joint ranking is further applied in inference time.

Results. The reuslts of our methods and the baseline approach on the ActivityNet Captions validation split are provided in Tab. 1. We can see that, our 6 variants all outperform the baseline method with large margins.

Compared to the baseline SST + H, our bidirectional model (Bi-SST + H) gives better performance when captioning 1000 proposals. This verifies that considering both past and future event context also help improve describing an event.

Combining both event clip features and context vectors (Bi-SST+E+H) is better than event clip features (Bi-SST+E) or context vectors (Bi-SST+H) alone. We mainly refer to Meteor score for comparison, as it shows better consistency with human judgments with a small number of reference sentences (in our case, only one reference sentence). We notice there is slight inconsistency for other metrics, which has also been observed by [45, 20]. This is caused by the imperfection of sentence similarity measurement.

Figure 4. Qualitative dense-captioning analysis for model without or with event clip fusion. Note that we only show proposals with maximum tIoU with the ground truths. (Best viewed in color)

Based on the results of Bi-SST+E+H+TDA, applying attention mechanism instead of mean pooling to dynamically fuse event clip features and context vectors further improves all scores. This variant performs better as it can generates more semantic related word by attending on video features at each decoding step. Combining context gating function further boosts the performance with clear margins. This supports that explicitly modeling the relative contribution from event features and contexts in decoding time help better describe the event. Using joint ranking at inference time further improve the whole system, as it gives more confident predictions on both event proposals and corresponding descriptions.

In Tab. 2, comparsion of our system with the state-of-the-art method [20] is presented. Note that our approach uses only C3D features and does not involve any extra data. While totally comparable to Krishna *et al.* [20], our method surpasses [20] with **100%** performance gain. This strongly supports the effectiveness of our proposed model.

Qualitative Analysis. For intuitively analyzing the effectiveness of fusing event clip for dense captioning, we show some cases in Fig. 4. The fusion mechanism allows the system to pay more "attention" to current event while simultaneously referring to contexts, and thus can generate more semantic-related sentences. In contrast, the system without event clip fusion generally tends to make more semantic mistakes, either incorrect (Fig. 4 (a) and (b)) or semantic ambiguous (Fig. 4 (c)). For example, when describing video (c), by incorporating event clip features, the system is more confident to say "The man is surfing with a surf board" instead of simply saying "riding down the river."

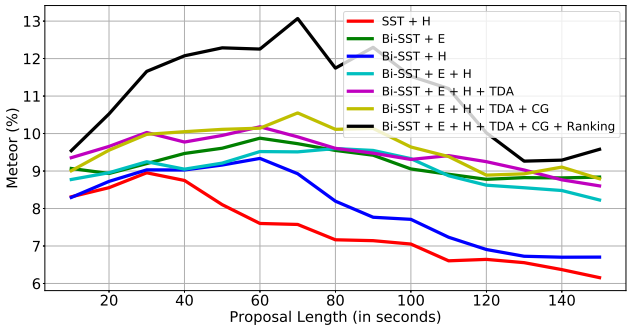


Figure 5. Meteor scores *vs* event proposal lengths.

Fig. 5 shows how Meteor scores change as proposal lengths vary from a few seconds to several minutes. We can see that the performances of all methods degenerate when describing very long proposals ($>$ 60s). This suggests that understanding long events and describing them is still a challenging problem, as long events are usually more complicated. Bi-SST+H works better than SST-H as we combine both past and future context information. We note that SST+H and Bi-SST+H both go down steeply as proposals become longer. The reason is that it is still very hard for LSTM to learn long-term dependency. Using only hidden states to represent an event is thus quite suboptimal. In contrast, fusing event features compensates such information loss. All methods using "E" (event features) show much better performance than their counterparts. Besides, our model with joint ranking further improves the performance of the whole system with large margins.

5. Conclusion

In this paper we identified and handled two challenges on the task of dense video captioning, which are context fusion and event representation. We proposed a novel bidirectional proposal framework, namely, Bidirectional SST, to encode both past and future contexts, with the motivation that both past and future contexts help better localize the current event. Building on this proposal module, we further reused the proposal hidden states as context vectors and dynamically fused with event clip features to generate the visual representation. The extensive quantitative and qualitative experimental results demonstrate the superiority of our model in both localizing events and describing them.

Acknowledgement The author Yong Xu would like to thank the supports by National Nature Science Foundation of China (U1611461 and 61672241), and the Cultivation Project of Major Basic Research of NSF-Guangdong Province (2016A030308013).

References

[1] L. Anne Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell. Localizing moments in video with natural language. In *ICCV*, 2017. 1

[2] S. Banerjee and A. Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005. 7

[3] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. Carlos Niebles. SST: Single-stream temporal action proposals. In *CVPR*, 2017. 1, 2, 5, 6, 7

[4] F. Caba Heilbron, J. Carlos Niebles, and B. Ghanem. Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In *CVPR*, 2016. 2

[5] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 6

[6] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *CVPR*, 2017. 1

[7] S. Chen, J. Chen, and Q. Jin. Generating video descriptions with topic guidance. In *ICMR*, 2017. 2, 3

[8] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem. DAPs: Deep action proposals for action understanding. In *ECCV*, 2016. 2, 3

[9] C. Gan, N. Wang, Y. Yang, D.-Y. Yeung, and A. G. Hauptmann. Devnet: A deep event network for multimedia event detection and evidence recounting. In *CVPR*, 2015. 1

[10] 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. 2, 3

[11] J. Gao, C. Sun, Z. Yang, and R. Nevatia. Tall: Temporal activity localization via language query. *arXiv preprint arXiv:1705.02101*, 2017. 1

[12] J. Gao, Z. Yang, K. Chen, C. Sun, and R. Nevatia. TURN TAP: Temporal unit regression network for temporal action proposals. In *ICCV*, 2017. 2

[13] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *ICCV*, 2013. 2

[14] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 1997. 2, 4

[15] W. Jiang, L. Ma, X. Chen, H. Zhang, and W. Liu. Learning to guide decoding for image captioning. In *AAAI*, 2018. 1

[16] Y. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. Thumos challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/, 2014. 6

[17] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 4

[18] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[19] R. Kiros, R. Salakhutdinov, and R. Zemel. Multimodal neural language models. In *ICML*, 2014. 2

[20] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. Carlos Niebles. Dense-captioning events in videos. In *ICCV*, 2017. 1, 2, 3, 4, 6, 7, 8

[21] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *WAS*, 2004. 7

[22] X. Long, C. Gan, and G. de Melo. Video captioning with multi-faceted attention. *Transactions of the Association for Computational Linguistics*, 2018. 2

[23] Z. Lu and K. Grauman. Story-driven summarization for egocentric video. In *CVPR*, 2013. 1

[24] L. Ma, Z. Lu, and H. Li. Learning to answer questions from image using convolutional neural network. In *AAAI*, 2016. 1

[25] L. Ma, Z. Lu, L. Shang, and H. Li. Multimodal convolutional neural networks for matching image and sentence. In *ICCV*, 2015. 1

[26] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, 2016. 2, 3

[27] Y. Pan, T. Yao, H. Li, and T. Mei. Video captioning with transferred semantic attributes. In *CVPR*, 2017. 2, 3

[28] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 7

[29] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. In *ECCV*, 2014. 1

[30] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang. Single image dehazing via multi-scale convolutional neural networks. In *ECCV*, 2016. 1

[31] W. Ren, J. Pan, X. Cao, and M.-H. Yang. Video deblurring via semantic segmentation and pixel-wise non-linear kernel. In *ICCV*, 2017. 1

[32] A. Rohrbach, M. Rohrbach, W. Qiu, A. Friedrich, M. Pinkal, and B. Schiele. Coherent multi-sentence video description with variable level of detail. In *GCPR*, 2014. 3

[33] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele. Translating video content to natural language descriptions. In *ICCV*, 2013. 2, 3

[34] Z. Shou, D. Wang, and S.-F. Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, 2016. 2

[35] C. G. Snoek and M. Worring. Concept-based video retrieval. *Foundations and Trends in Information Retrieval*, 2008. 1

[36] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 2, 4

[37] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015. 7

[38] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence-video to text. In *ICCV*, 2015. 2, 3

[39] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015. 5

[40] J. Wang, J. Fu, Y. Xu, and T. Mei. Beyond object recognition: Visual sentiment analysis with deep coupled adjective and noun neural networks. In *IJCAI*, 2016. 1

[41] J. Xu, T. Mei, T. Yao, and Y. Rui. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*, 2016. 1

[42] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICLR*, 2015. 3, 5

[43] Y. Xu, H. Ji, and C. Fermüller. Viewpoint invariant texture description using fractal analysis. *IJCV*, 2009. 1

[44] H. Yang and C. Meinel. Content based lecture video retrieval using speech and video text information. *IEEE Transactions on Learning Technologies*, 2014. 1

[45] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. In *ICCV*, 2015. 2, 3, 5, 7

[46] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei. Boosting image captioning with attributes. In *ICCV*, 2017. 3

[47] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *CVPR*, 2016. 3

[48] Y. Yuan, X. Liang, X. Wang, D.-Y. Yeung, and A. Gupta. Temporal dynamic graph lstm for action-driven video object detection. In *ICCV*, 2017. 1

[49] X. Zhang, K. Gao, Y. Zhang, D. Zhang, J. Li, and Q. Tian. Task-driven dynamic fusion: Reducing ambiguity in video description. In *CVPR*, 2017. 2, 3