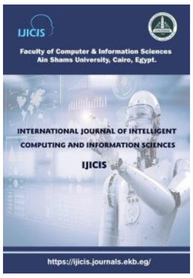
IJICIS, Vol.21, No.2, 1-22
DOI: 10.21608/ijicis.2021.60216.1055

International Journal of Intelligent Computing and Information Sciences

https://ijicis.journals.ekb.eg/


IJICIS
Faculty of Computer & Information Sciences Ain Shams University, Cairo, Egypt.
INTERNATIONAL JOURNAL OF INTELLIGENT COMPUTING AND INFORMATION SCIENCES
IJICIS
https://ijicis.journals.ekb.eg/

Bidirectional Temporal Context Fusion with Bi-Modal Semantic Features using a gating mechanism for Dense Video Captioning.

Noorhan K. Fawzy
Computer Science Department, Faculty of Computer and Information Sciences, Ain Shames University, Cairo, Egypt
norhan.khaled@cis.asu.edu.eg

Mohammed A. Marey
Scientific Computing Department,
Faculty of Computer and Information Sciences, Ain Shames University, Cairo, Egypt
Mohammed.marey *@cis.asu.edu.eg*

Mostafa M. Aref
Computer Science Department, Faculty of Computer and Information Sciences, Ain Shames University, Cairo, Egypt
mostafa.aref *@cis.asu.edu.eg*

Received 2021- 1-28; Revised 2021-7-11; Accepted 2021-7-13

Abstract: *Dense video captioning involves detecting interesting events and generating textual descriptions for each event in an untrimmed video. Many machine intelligent applications such as video summarization, search and retrieval, automatic video subtitling for supporting blind disabled people, benefit from automated dense captions generator. Most recent works attempted to make use of an encoder-decoder neural network framework which employs a 3D-CNN as an encoder for representing a detected event frames, and an RNN as a decoder for caption generation. They follow an attention based mechanism to learn where to focus in the encoded video frames during caption generation. Although the attention-based approaches have achieved excellent results, they directly link visual features to textual captions and ignore the rich intermediate/high-level video concepts such as people, objects, scenes, and actions. In this paper, we firstly propose to obtain a better event representation that discriminates between events nearly ending at the same time by applying an attention based fusion. Where hidden states from a bi-directional LSTM sequence video encoder, which encodes past and future surrounding context information of a detected event are fused along with its visual (R3D) features. Secondly, we propose to explicitly extract bi-modal semantic concepts (nouns and verbs) from a detected event segment and equilibrate the contributions from the proposed event representation and the semantic concepts dynamically using a gating mechanism while captioning. Experimental results*

* Corresponding author: Noorhan K. Fawzy
Computer Science Department, Faculty of Computer and Information Sciences, Ain Shames University, Cairo, Egypt
E-mail address: norhan.khaled@cis.asu.edu.eg

demonstrates that our proposed attention based fusion is better in representing an event for captioning. Also involving semantic concepts improves captioning performance.

Keywords: *Video events proposal detection, Video to natural language, Attention-Based sentence decoder, Bidirectional LSTM, Deep Learning*

1. Introduction

Describing what happened within a brief video using a natural language sentence is referred to video captioning [1]. This is an easy task for humans but for machines, the need for learning relevant semantics from raw pixels for generating natural words is complex. Also natural videos are long and may contain multiple events, such that for example, at some point in the video, a man was shooting bullets towards a target, later on another man hits him from his back, after that the target ran, Fig.1.

Fig. 1: Example of multiple events in an untrimmed video.

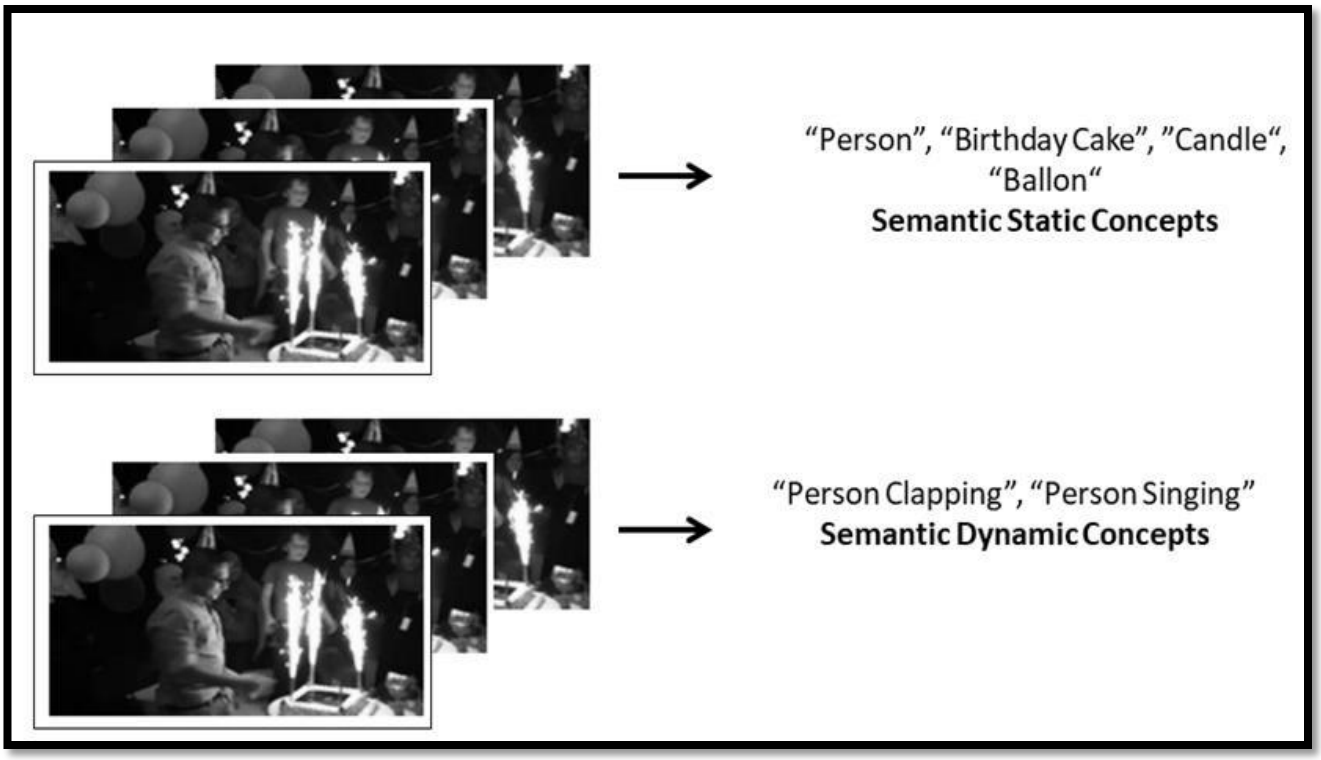

“Person”, “Birthday Cake”, ”Candle“, “Ballon“
Semantic Static Concepts
“Person Clapping”, “Person Singing”
Semantic Dynamic Concepts

Fig. 2: Examples of semantic static (nouns) and dynamic (verbs) concepts.

For this, a single sentence can’t capture the semantics complexity of the whole scenario within the video. Therefore, recently the focus has been shifted towards generating multiple sentences and relating them with time locations which is referred to dense captioning. The need for both localizing temporal event segments boundaries occupied within the video and generating a sentence per event must be considered. It is important to note that, the direct linking of visual features from event segments to textual natural language sentences may ignore the rich intermediate/high-level descriptions, such as objects, scenes, and actions, ex: (a video of birthday party contains semantic static concepts (nouns) such as “a person” , “birthday cake” , ” candle “, “ballon “ and semantic dynamic concepts (verbs) such as “clapping” , “waving” , ⋯ etc), as illustrated in Fig. 2. For this, it is crucial to identify semantic features from the input video frames for implementing an effective caption generation process.

A broad range of applications can benefit from the localization and description of events in videos, such as video summarization, search and retrieval, automatic video subtitling for supporting blind disabled people.

Recently many works [1-3] attempted to make use of an encoder-decoder neural network framework which employs a 3D-CNN as an encoder for representing a detected event frames, and an RNN as a decoder for caption generation. They formulate the dense captioning task as a sequence-to-sequence (video to caption) task.

Some subsequent works [3-6] follow an attention based mechanism to learn where to focus in the encoded video frames during caption generation. Although the attention-based approaches have achieved excellent results, they directly link visual features to textual captions and ignore the rich intermediate/high-level video concepts such as people, objects, scenes, and actions.

In this work we propose a deep neural network framework that firstly utilizes a bi-directional LSTM network which encodes past and future local contextual information while detecting human action event segments. It applies an attention based fusion of hidden states from the bi-directional LSTM sequence video encoder along with its visual (R3D) features, to obtain a better event representation that discriminates between events nearly ending at the same time. Secondly, we propose to explicitly extract bi-modal semantic concepts (nouns and verbs) from a detected event segment and equilibrate the contributions from the proposed event representation and the semantic concepts dynamically using a gating mechanism while captioning.

In section 2 we will discuss the related works, section 3 will introduce the proposed framework in details. Experiments and implementation details will be provided in section 4. Finally the conclusion will be discussed in section 5.

2. Related Works

Dense video captioning requires localizing events in the input video and then produce a textual sentence description for each event happening in the video. The dense video captioning task branches out from the video captioning task which aims to caption a pre-segmented video without the need to localize the event.

Untrimmed videos contain target events that usually occupy a small portion of the whole video stream. Some current methods for action temporal localization [7, 8] rely on applying action classifiers at every time location and at multiple temporal scales, in a temporal sliding window fashion. Major drawbacks regarding these methods are, the high computational complexity, scales of sliding windows is predetermined based on the statistics of the dataset and temporal boundaries generated are usually approximate and fixed during classification.

Recent researches in dense video captioning [9, 10] worked on avoiding the high computational complexity drawbacks of sliding windows. They followed an end to end event proposal generation model that produces confidence scores of multiple event proposals with multiple time scales at each time step of the video. They process each input frame only once and thereby process the full video in a single pass. They argued that a successful proposal generation method should be able to retrieve a small number of event intervals with high recall and high temporal overlap with true event segments. Their method mainly scans an untrimmed video stream of length L frames divided into $T=L/\delta$ non-overlapping time steps, where $\delta=16$ frame. Each time step is encoded with the activations from the top layer of a 3D convolutional network pre-trained for action classification (C3D network [11]). A recurrent neural network (RNN) was used for modeling the sequential information into a discriminative sequence of hidden states. The hidden representation at each time step is used for producing confidence scores of multiple proposals with multiple time scales that all end at time t. However, these methods simply neglects future event context and only encode past and current event context information when predicting proposals.

After obtaining events segments, early approaches in deep learning directly connected event frames with language. Translating video pixels to natural language was their aim. Inspired by the successful use of the encoder-decoder framework employed in machine translation, many existing works on video captioning [1, 2] employ a convolutional neural network (CNN) like ResNet [12], C3D [11] or two-stream network [13] as an encoder, obtaining a fixed-length vector representation of a given video. A mean pooling of features across all frames is applied for obtaining a fixed-length vector representation, which is considered as a simple and reasonable semantic representation for short video clips. Translation to natural language is done via a stacked two-layer recurrent neural network (RNN), typically implemented with long short-term memory (LSTM) [14, 15]. However, they considered frame features of the video equally, without any particular focus. Using a single temporally collapsed feature vector for representing such videos leads to the incoherent fusion of the dependencies and the ordering of activities within an event.

Some subsequent methods [3-6] explored strengthening the semantic relationship between a video and the corresponding generated words of a sentence by including attention models in video context. They took inspiration from the soft attention mechanism [16] applied in image captioning. They argued that videos consist of spatial (frame-level) features and their temporal evolutions, an effective captioning model should be able to attend to these different cues selectively for selecting significant regions in the most relevant temporal segments for word prediction.

Yao et al [3] proposed a novel spatiotemporal 3D CNN, which accepts a 3-D spatio-temporal grid of cuboids. These cuboids encode the histograms of oriented gradients, oriented flow and motion boundary (HoG, HoF, and MbH) [17]. They argued that, average pooling these local temporal motion features would collapse and neglect the model's ability to utilize the video's global temporal structure. For this a soft attention mechanism was adapted, which permits the RNN decoder weighting each temporal feature vector when predicting next word.

.

Also the work in [5] proposes the use of a novel spatial-temporal attention mechanism (STAT) within an encoder-decoder neural network for video captioning. Firstly, they use 2-D/3-D Convolutional Neural Network (CNN) and Region-based Convolutional Neural Networks (RCNNs) to encode the video inputs to a set of fixed length vector representation. Secondly, they use three kinds of features via two-stage attention mechanism. In first stage a spatial attention mechanism makes the decoder to select local features with more spatial attention weights, which represent the significant regions. Then in second stage a temporal attention mechanism make the decoder to select global and motion features, as well as local features representing significant regions. Finally, three types of features are fused to represent the information of key frames.

[18] Propose to generate event proposal first and then dynamically select (attend) neighbouring events as context for target event captioning. They categorize all events into two buckets (past and future) relative to a current event. They trained their model to apply an attention mechanism that adequately focuses on selecting, events that have already occurred (past), and events that take place afterwards (futures), which are relevant to the current event.

Although the attention-based approaches have achieved excellent results, they still ignore representing high-level video concepts/attributes. Being aware of linguistic contexts of an event not only can provide holistic information to understand the event more accurately, but also tell differences between target event and its context to generate more diverse captions.

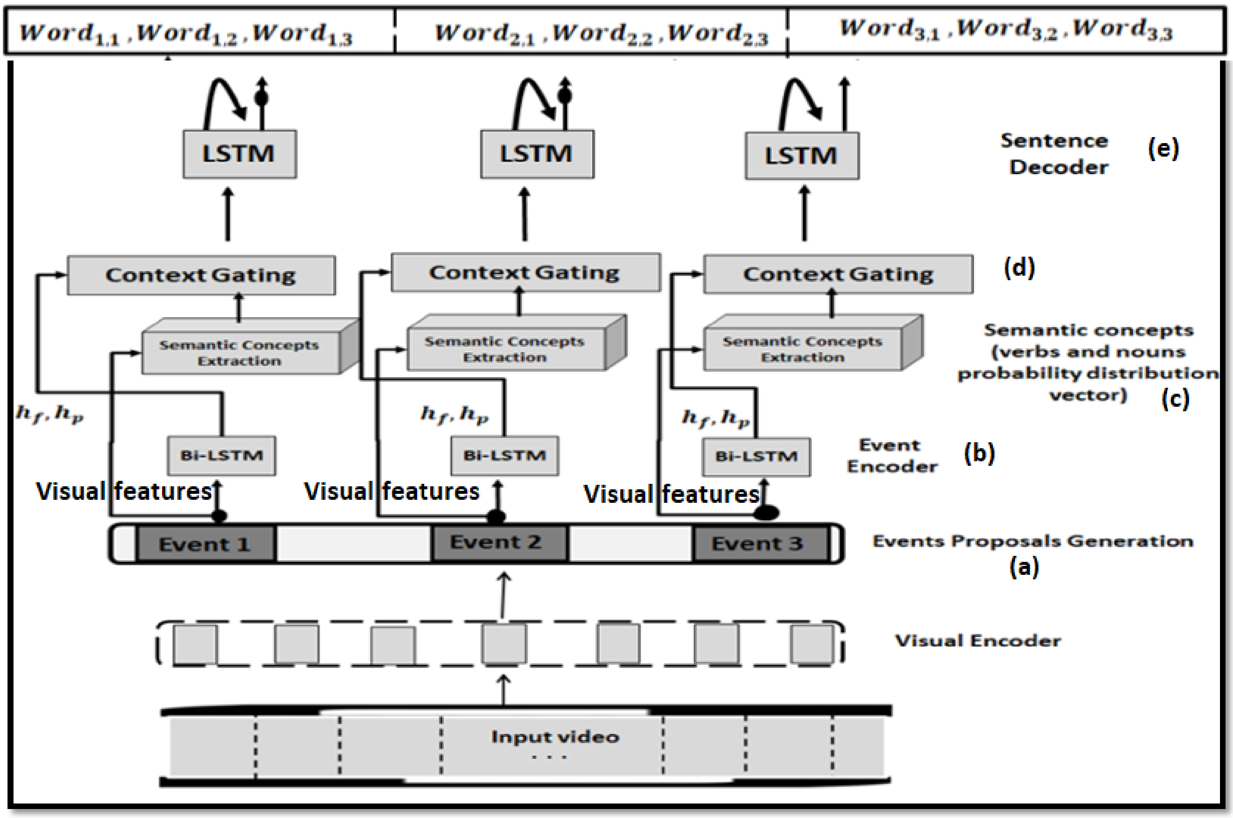

$Word_{1,1}, Word_{1,2}, Word_{1,3}$
$Word_{2,1}, Word_{2,2}, Word_{2,3}$
$Word_{3,1}, Word_{3,2}, Word_{3,3}$
LSTM
LSTM
LSTM
Sentence Decoder
(e)
Context Gating
Context Gating
Context Gating
(d)
Semantic Concepts Extraction
Semantic Concepts Extraction
Semantic Concepts Extraction
Semantic concepts (verbs and nouns probability distribution vector)
(c)
h_f, h_p
h_f, h_p
h_f, h_p
Bi-LSTM
Bi-LSTM
Bi-LSTM
Event Encoder
(b)
Visual features
Visual features
Visual features
Event 1
Event 2
Event 3
Events Proposals Generation
(a)
Visual Encoder
Input video
. . .

Fig. 3: The architecture of the framework for dense event captioning.

3. Proposed Method

In this section we will introduce our adopted method for densely describing events in videos. Mainly our method is based upon two stages: 1) event proposal generation for obtaining a set of potential candidate temporal regions that contains possible events, Fig. 3 (a). 2) Caption generation for producing a sentence per event segment, Fig. 3 (e).

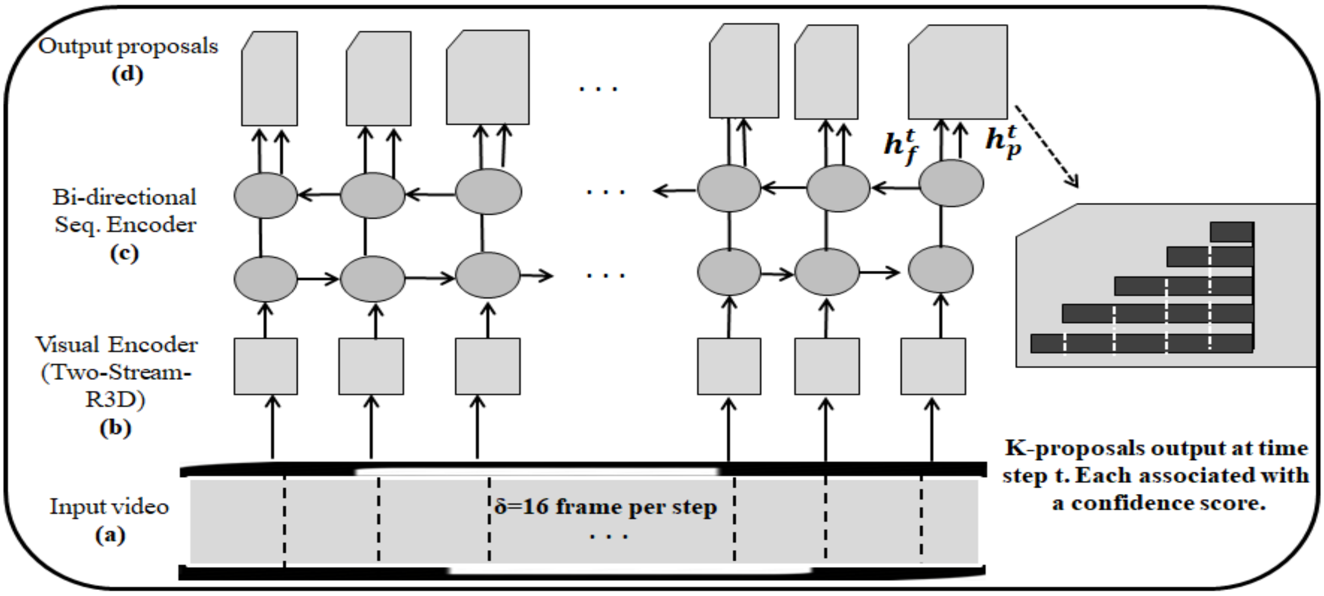


Fig. 4: The bi-directional LSTM sequence encoder that can represent a set of K temporal intervals at each time step using forward and backward hidden states.

Both stages are coupled together in a deep learning framework that follows the encoder-decoder model. Each stage will be discussed in details in the following sections. The overall framework is illustrated in Fig. 3.

3.1. Event Proposals generation

In this stage, we work on obtaining a set of candidate temporal regions that contains potential events, as illustrated in Fig. 3 (a). First, given a video with L frames, we discretize it by dividing into T non-overlapping time steps, where each time step is of size δ =16-frames, Fig. 4 (a). We used a two-stream 3D Residual Neural Network for the extraction of spatio-temporal features from each time step, Fig.4 (b). One stream for learning to extract motion features from RGB frames using 3D ResNet-18 [12]. The other for learning abstract high level motion features from motion boundary frames using 3D ResNeXt-101 [12]. Motion boundary frames carry optimized smooth optical flow inputs.

We propose to adopt a bidirectional recurrent neural network model, especially LSTM (long short term memory) [14], which produce two hidden states representing past and future context information for sequentially encoding the visual features of the video time steps, Fig. 4 (c). The input video visual features are fed to the forward LSTM in its neutral order and to the backward LSTM in a reversed order. The LSTM outputs two hidden state values, $\mathrm{h}_\mathrm{f}^\mathrm{t}$ and $\mathrm{h}_\mathrm{p}^\mathrm{t}$ where $\mathrm{h}_\mathrm{f}^\mathrm{t}$ is generated from the forward pass and $\mathrm{h}_\mathrm{p}^\mathrm{t}$ is generated from the backward pass. They encode visual information observed from past and future time steps at time step (t) respectively.

At each time step (t), we pass the hidden state that encodes the sequence of visual features observed till time (t) through a fully connected layer with sigmoid nonlinear function 6, as in Eq. (1) and Eq. (2), for producing multiple (K) proposals scores, as Fig. 4 (d). Proposals have different time scales with a fixed ending boundary (t) and (K) confidence scores. These scores indicate the probabilities of K proposals, each specified with a number of time steps denoted by $\mathrm{k}^{\rightarrow t}= \{\mathrm{k}_{i}^{\rightarrow t}\}_{i=1}^{K}$. $\mathrm{k}_{i}^{\rightarrow t}$, denotes a video temporal segment (proposal) with end time as t and start time as $\mathrm{t}-\mathrm{l}_{i}$ as illustrated in Fig. 5. Where $\{\mathrm{l}_{i}\}_{i=1}^{K}$is the lengths of the predefined K proposal anchors.

This is done for each LSTM direction (forward backward) independently. Finally the passes through the two directions, generates N proposals collected from all time steps of both directions. Many overlapping proposals can be generated by a single iteration over the input video using a LSTM. For selecting highly confident proposals, we fuse the two sets of scores for the same proposals, yielding the final scores, through multiplication, as in Eq. (3). A predicted event proposal (i) is associated with a start time, an end time, a confidence score, and two hidden states $\boldsymbol{h_f^t}$ and $\boldsymbol{h_p^t}$ as illustrated in Eq. (4).

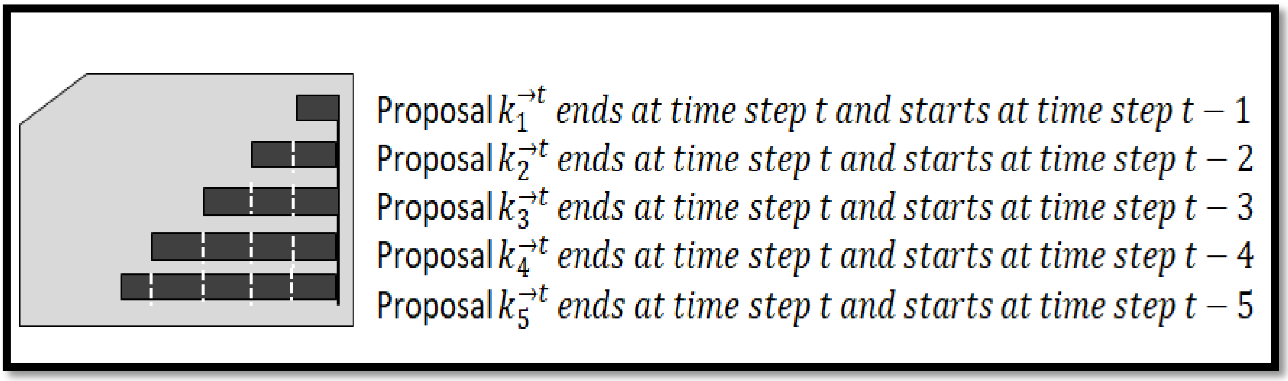
$$\boldsymbol{C_i^{\rightarrow}} = 6\left(\boldsymbol{W_i . h_f^t}\right), \text{i= \{1... k\}} \tag{1}$$

$$\boldsymbol{C_i^{\leftarrow}} = 6\left(\boldsymbol{W_i . h_p^t}\right), \text{i= \{1... k\}} \tag{2}$$

$$\boldsymbol{C_i^t} = \{\, \boldsymbol{C_i^{\rightarrow}\ X\ C_i^{\leftarrow}} \,\}_{\boldsymbol{i=1}}^{\boldsymbol{k}} \tag{3}$$

$$\boldsymbol{p^i} = \{(\boldsymbol{t_{start}^i}, \boldsymbol{t_{end}^i}, \boldsymbol{Score^i}, \boldsymbol{h_f^t}, \boldsymbol{h_p^t})\} \tag{4}$$

The summarized algorithmic steps can be illustrated in Fig. 6.



Proposal $k_1^{\rightarrow t}$ *ends at time step t and starts at time step t* − 1
Proposal $k_2^{\rightarrow t}$ *ends at time step t and starts at time step t* − 2
Proposal $k_3^{\rightarrow t}$ *ends at time step t and starts at time step t* − 3
Proposal $k_4^{\rightarrow t}$ *ends at time step t and starts at time step t* − 4
Proposal $k_5^{\rightarrow t}$ *ends at time step t and starts at time step t* − 5

Fig. 1: Multiple proposals (temporal segments) with different time scales that all end at time step t.

Input: A video stream of L frames X= {x_1,x_2,…,x_L} that is discretized into T non overlapping (T=L/ δ) time steps (clips), $\hat{X}$ ={t_1,t_2,…,t_T} each time step (clip) of resolution δ=16 frame.

Step 1: For each frame i in time step t of resolution δ
Compute a feature representation v_t that encapsulates the visual content of the frames within each time step t, through the two-stream 3D Residual neural network (R3D), v_t=$two\_stream\ R3D(\{t^{frame}\}_{frame=ix\delta}^{(ix\ \delta)+\delta}$
End For

Step 2: Perform PCA to reduce the dimensionality of each visual feature representation v_{t_i}.

Step 3: Pass the corresponding visual feature representation v_t of each time stept to the bi-directional LSTM (bi-LSTM) sequence encoder for encoding the stream of visual features and producing a forward hidden state value h_f^t and a backward hidden state value h_p^t for each v_t .

Step 4: Apply K independent binary classifiers (Action-Background) representing K proposals (temporal segments) with K different time scales that all end at t_i, on top of each LSTM direction. The classifier is represented by a fully connected layer with sigmoid nonlinear function. At each time stept, the classifier on top of the forward LSTM layer accepts the forward hidden state value h_f^t and the classifier on top of the backward LSTM accepts the backward hidden state value h_p^t. This produces forward confidence score $C_i^{\rightarrow}$ and backward confidence score $C_i^{\leftarrow}$ of multiple proposals at each time step t_i.

Step 5: Multiply the confidence score of the kth proposal produced from each direction (forward $C_i^{\rightarrow}$x backward $C_i^{\leftarrow}$) to get the final confidence score C_p^t.

Output: The proposals at time step t_i with a score larger than a threshold are selected as action proposals for further captioning. Each action proposal has a start time step S and an end time step E, confidence score and a hidden state from both forward and backward directions, p^i = $\{(t_{start}^i, t_{end}^i, Score^i, h_f^t, h_p^t)\}$

Fig. 2: Summarized algorithmic steps of the event proposal generation module.

3.2. Caption Generation

Once proposals are obtained, a LSTM network decoder translates the encoded sequence of a detect event clips visual inputs to a sentence, Fig. 3 (e).

It is important to mention that, contextual information plays an important role in the task of dense video captioning. Contextual reasoning is vital for providing holistic information to understand an event more accurately and generating more diverse captions.

In this section, we investigate different contexts for encoding detected event proposals for dense captioning. Two types of context are utilized during captioning, including:

- **Local visual contextual information:** Modeling the contents that are temporally neighboring the target event proposal which are encoded in the proposal's start 'S' and end 'E' time step hidden states (h_f^S,h_p^E) of the forward and backward LSTM, and is dynamically fused with the event's visual features via an attention based mechanism, as in Eq. (5). It is not preferred to discard worthy information encoded in future and past neighboring regions of the video. These local neighboring regions provide significant precedents and consequences for understanding an event.

$$r_i(S_n) = \text{fusion}([h_f^S, h_p^E], V, H_{t-1}), n \in \{1 \dots N\}, i \in \{S \dots E\} \quad (5)$$

Where V= $\{v^i\}_{i=S}^E$ is the two-stream visual feature maps from the start clip S and end clip E of the n^{th} event proposal S_n. (h_f^S, h_p^E) Are the local visual context vectors of the n^{th} event proposal. This process can be illustrated in details through Fig. 3 7.

The unnormalized relevance score r_i for the i^{th} clip's features (V^i) within a proposal, can be obtained through a weighted linear combination as Eq. (6) using a neural network layer with hyperbolic tangent (tanh) activation function. Where, S and E denote the start and end time steps of the proposal. H_{t-1} Is the hidden state of the sequence encoder at the t-1 time step. Vector concatenation is applied on (h_f^S,h_p^E), represented by the [,] operator. The weights β^i of each visual feature V^i can be obtained by softmax normalization (that converts a neural network layer's output into a probability distribution) such as Eq. (7). The final attended visual features $\widehat{V}$ are generated by a weighted sum through Eq. (8).

$$\boldsymbol{fusion = W_a^T . tanh\ (W_v V^i + W_h [h_f^S, h_p^E}\ ] + \boldsymbol{W_H H_{t-1} + b}\ ), i \in \{S \dots E\} \quad (6)$$

$$\boldsymbol{\beta^i} = \exp\ (\boldsymbol{r_i})\ / \textstyle\sum_{m=S}^{E} \boldsymbol{r_m} \quad (7)$$

$$\boldsymbol{\widehat{V} = \textstyle\sum_{i=S}^{E} \beta^i . V^i} \quad (8)$$

$$\boldsymbol{input_t} = [\boldsymbol{h_f^S}, \boldsymbol{h_p^E}, \boldsymbol{\widehat{V}}\ ] \quad (9)$$

This attended visual features $\widehat{V}$ can be concatenated with context vectors (h_f^S,h_p^E), represented by the [,] operator, to produce visual_context_n vector of the n^{th} event proposal, as illustrated in Eq. (9). This vector can be passed through the gates of the decoding LSTM.

This is done to produce a robust event representation that can help discriminate between events that nearly end at the same time, since the detected temporal neighboring regions will be different.

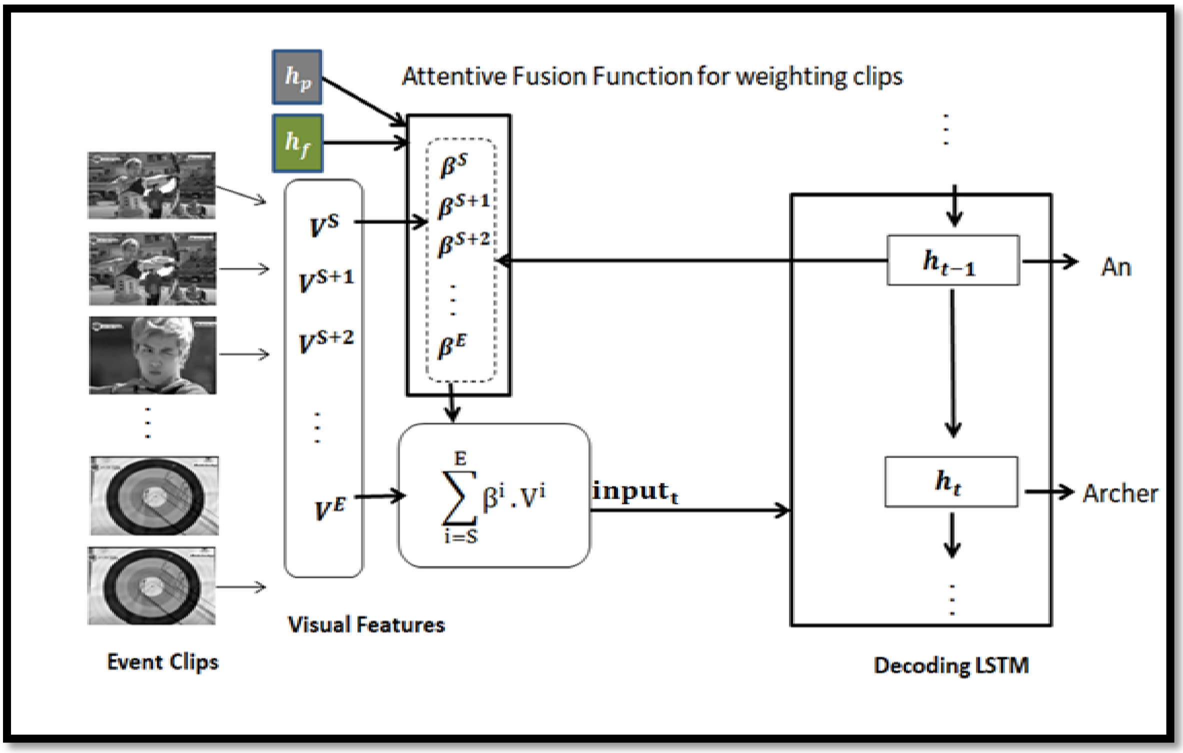


Fig. 3: Dynamic attention mechanism to fuse visual features and context vectors using an attention mechanism, at each time step while decoding.

- **Proposal-level semantic context:** which includes an explicit bi-modal (static and dynamic) semantic concept features extractor networks that predict significant nouns and verbs from video clips of an event, Fig. 3 (c). Concepts that describe objects, backgrounds/scenes are referred to static semantic concepts (nouns), Fig. 2. Concepts that describe dynamic actions are referred to dynamic semantic concepts (verbs), Fig. 2. These concepts are important and affect the contribution to the sentence generation. Semantic features are obtained using a LSTM network, in our framework, Fig.8 and Fig. 9. Indeed, obvious difference exists between these semantic features, so the extraction of these bi-modal features was done separately in the form of multi-label classification.

Each network outputs a probability distribution, where ($\mathrm{p_d}$) is the probabilities of the set of dynamic concepts (verbs) extracted from a clip at time step t of an input event proposal and ($\mathrm{p_s}$) is the probabilities of the set of static concepts (nouns). We consider the probabilities (of nouns and verbs) obtained from each time step t within the start S and End E of an input event proposal, as the extracted dynamic and static semantic concepts. It is important to note that the set of semantic static and dynamic concepts (nouns and verbs) are pre-defined from the dataset used while training this module.

This is done for reducing the fissure between low-level video features and sentence descriptions.

Both the probability distributions of the set of dynamic and static semantic concepts, that are extracted from the $\mathrm{n^{th}}$ event proposal are concatenated, resulting in $\mathrm{E_t} = \{\mathrm{E^i}\}_{\mathrm{i=1}}^{\mathrm{x}}$ (where x is the dimension of the concatenated set of concepts probabilities) and serve as input for the attention layer, as illustrated in Fig. 10. A weight value ($\mathrm{input_t} = \{\gamma^{\mathrm{i}}\}_{\mathrm{i=1}}^{\mathrm{x}}$) reflecting the semantic concept features to focus on at a time step t (of the decoding LSTM) is calculated by the attention layer for all the semantic features and illustrated in Fig. 10.

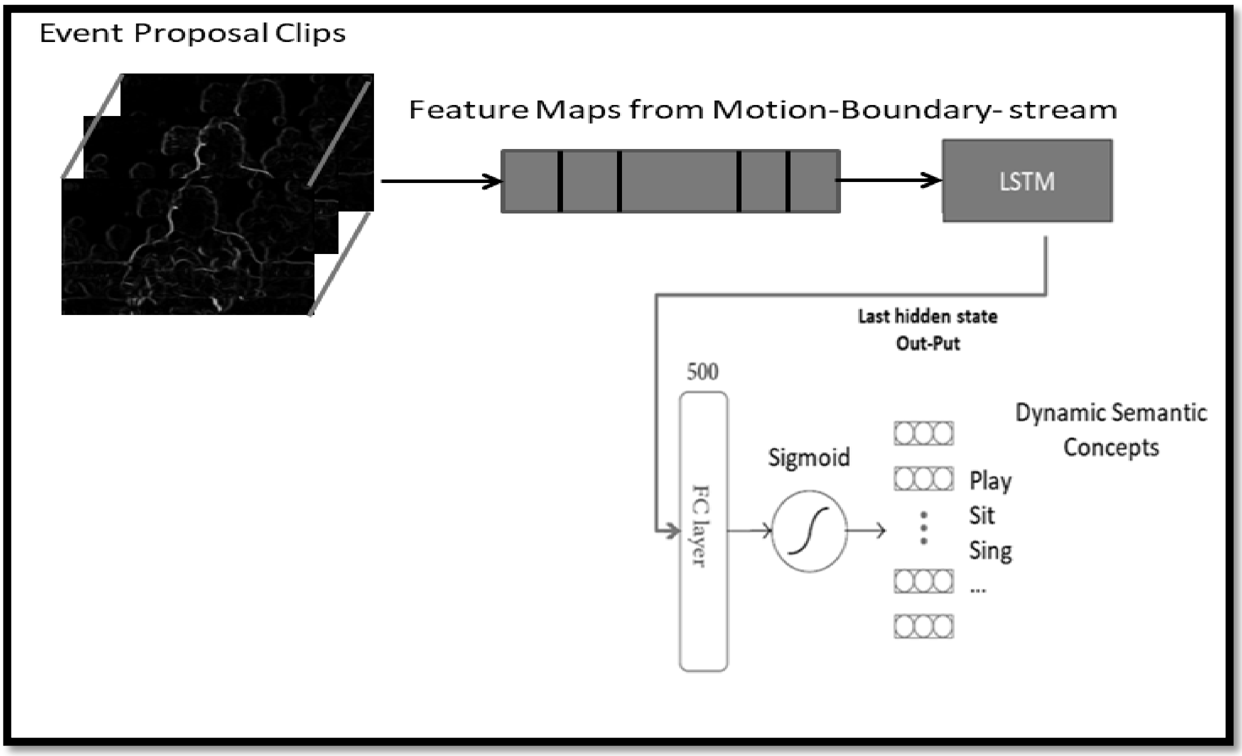


Fig. 8: Semantic dynamic concepts (verbs) extraction network

We balance the contributions from the proposed event representation and the semantic concepts dynamically using a gating mechanism while captioning.

Once obtain the attended visual feature vector $\widehat{\boldsymbol{V}}$, and the semantic context vector $\boldsymbol{\gamma_t}$we want to combine them together as input for the decoding LSTM. Inspired by the gating mechanism in LSTM, we develop a "context gate" to balance them. We propose to explicitly model the relative contributions of the attentive event visual feature vector, and semantic context vector when generating a word. The network is expected to determine how much context might be utilized when generating the next word.

In the context gating mechanism, the attended visual feature vector and the semantic context vector are projected into the same space, as Eq. 10 and Eq. 11.

$$\dot{\boldsymbol{V}}=\tanh\,(\widehat{\boldsymbol{W}}\widehat{\boldsymbol{V}}) \qquad (10)$$

$$\dot{\boldsymbol{\gamma}}=\tanh\,(\boldsymbol{W_\gamma \gamma}) \qquad (11)$$

The context gate is then calculated by a nonlinear layer as Eq.12.

$$\boldsymbol{G_{ctx}} = \boldsymbol{\sigma}\,(\boldsymbol{W_g}[\dot{\boldsymbol{V}},\dot{\boldsymbol{\gamma}},\boldsymbol{M_t},\boldsymbol{H_{t-1}}]\,) \qquad (12)$$

Where $\boldsymbol{M_t}$ is word embedding vector, $\boldsymbol{H_{t-1}}$ is the previous LSTM state. As, the context gate uses Sigmoid which outputs a value between 0 and 1, it can either let no flow or complete flow of information.

This gate models the contribution of each context information while decoding the event proposal. We then use the value of the context gate to fuse the attended event visual feature vector and the semantic context vector together, which is represented by $\boldsymbol{input_t}$ in Eq. 13 that is the input vector to the LSTM decoder units. This can be illustrated in Fig. 11.

$$\boldsymbol{input_t} = [(1-\boldsymbol{G_{ctx}}) \cdot \dot{\boldsymbol{V}}, \boldsymbol{G_{ctx}} \cdot \dot{\boldsymbol{\gamma}}, [\boldsymbol{h_f^{\rightarrow}}, \boldsymbol{h_b^{\leftarrow}}]] \qquad (13)$$

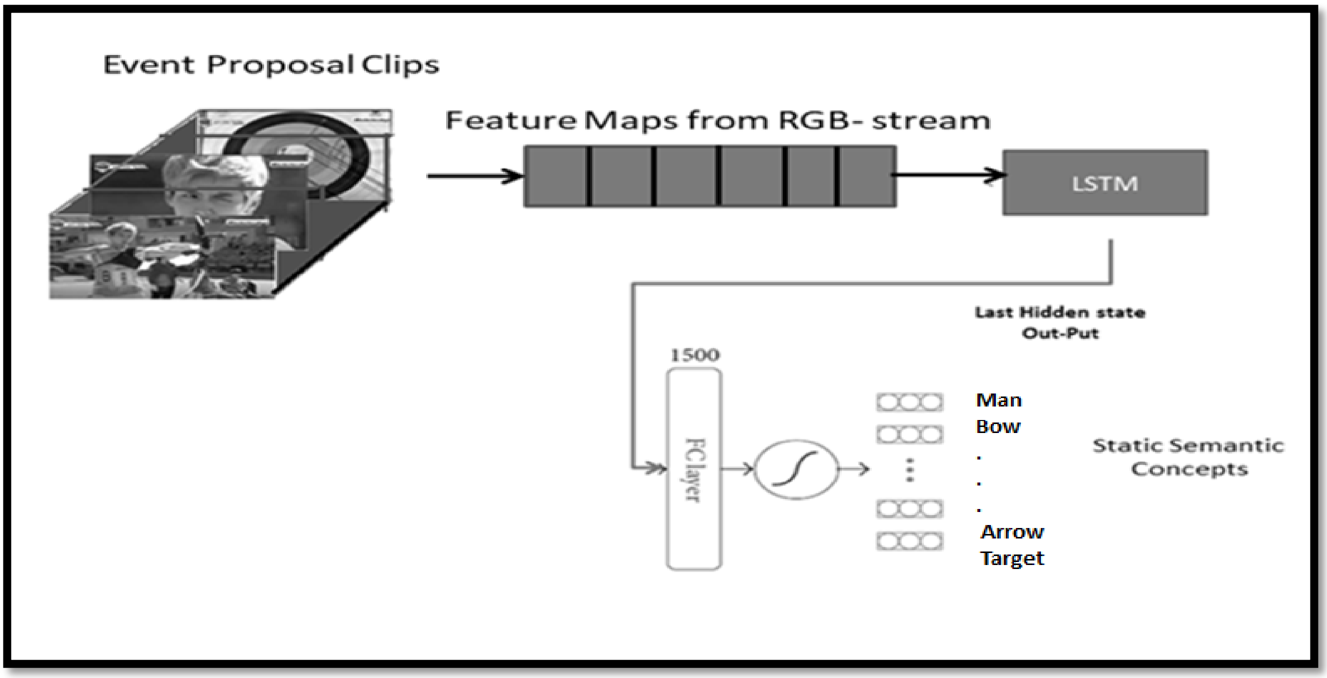


Table 1 Comparison of different variants for LSTM action proposal generation network.

Method	F1 Score
Bidirectional-Temporal proposal generation.	**0.807**
Single-direction Temporal proposal generation (Forward) [6]	0.74
Single-direction Temporal proposal generation (Backward)	0.71

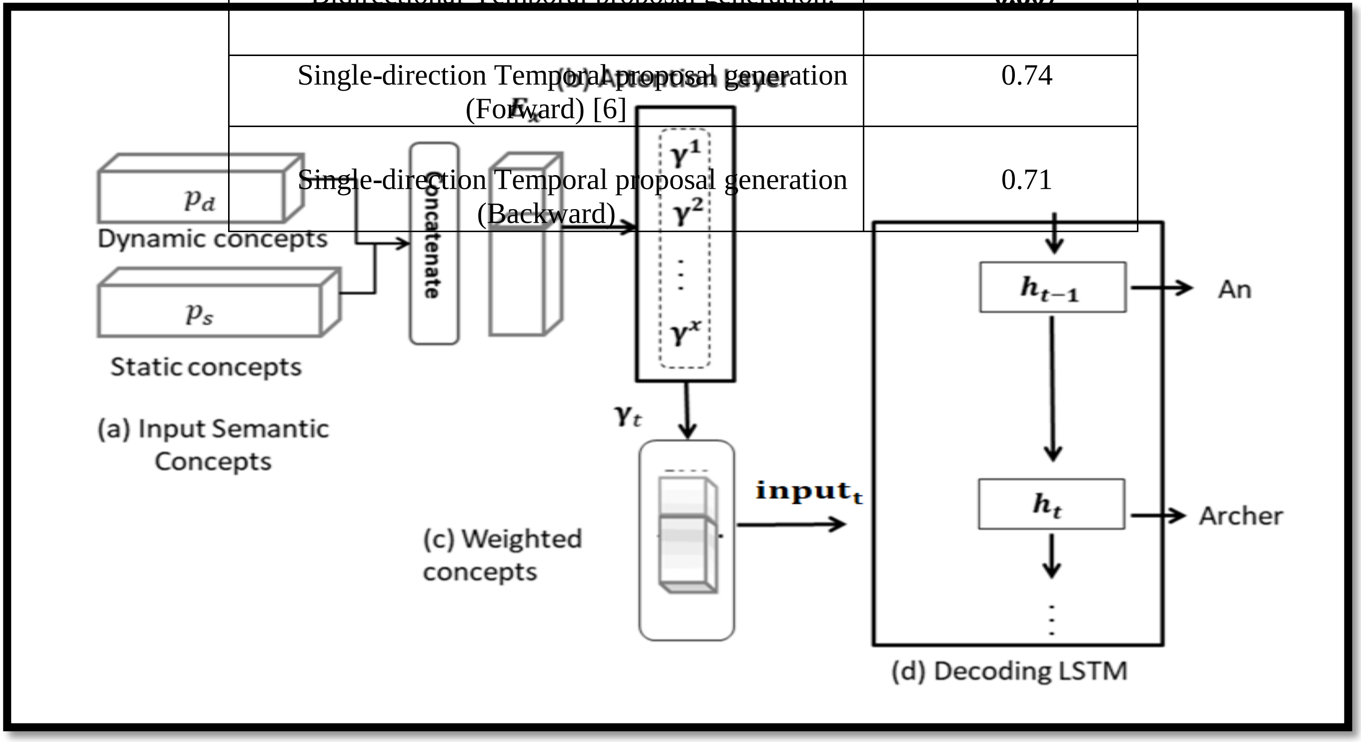


Fig. 5: Attention mechanism is applied for weighing the concatenated semantic concepts (dynamic and static) while decoding.

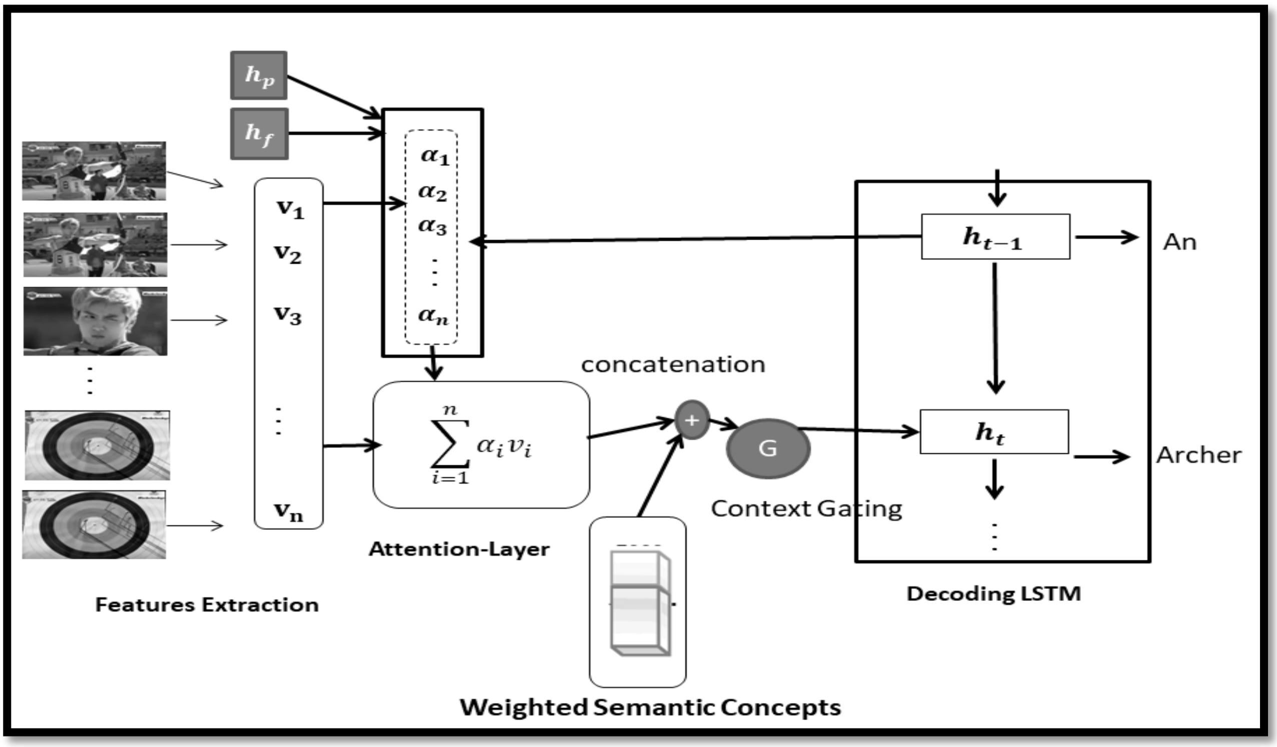


Fig. 4: Both the attended visual features and the weighted semantic concepts can be concatenated for balancing between them with the context gating mechanism.

4 Performance evaluation

4.1 Dataset

The ActivityNet Captions dataset [18] videos are bridged with a series of sentences that are annotated temporally. An event segment is described by a unique sentence. The videos from ActivityNet Captions dataset is the same as ActivityNet v1.3 dataset. It consists of 20,000 untrimmed videos from YouTube representing real life. The average videos length is roughly 120 seconds long. Mostly over 3 annotated events are contained within each video associated with start end time and human-written sentences. Activity Net includes activities in top level categories that are: House hold activities, sports and exercising, Personal care, Education and Working activities. In our experiments we only used videos from the “sports and exercising” category for training and evaluating our framework. The number of videos used in our experiments was 3,485 video that contains 48 different activity classes.

Each video contains 3.65 temporally localized sentences. We randomly split the data into three different subsets: train, validation and test, where 50% is used for training, and 25% for validation and testing. For ActivityNet Captions dataset, 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. Where longer strides are able to capture longer events. ActivityNet Captions focuses on verbs and actions.

4.2 Experiments

Event Proposals Generation

One of the main aims of this paper is developing models that are able to locate events within a video. Accordingly, we investigate how well our model can predict the temporal location of events within the corresponding video, in isolation of the captioning module. We compare our bidirectional proposal module with SST (Single Stream Temporal Proposal Generation) [9], which adopts a uni-direction

(Forward) LSTM model for proposal generation, using F1-score against tIoU threshold=0.8 with ground truth proposals.

Compared Methods

- F-TPG: Forward-Temporal Proposal Generation, used in [9].
- B-TPG: Backward-Temporal Proposal Generation.
- Bi-TPG: Bi-directional-Temporal Proposal Generation, which is the adopted method. We combine Forward and Backward LSTMs and jointly inference by fusing scores for the same predicted proposals.

The results recorded in Table 1 confirms that detections done with bidirectional model that encodes past and future context indeed upgrades the quality of the proposals, compared to a single direction prediction model. F1 score is a metric utilized to simultaneously consider both precision and recall for event localization, where it can be considered as a weighted average of the precision and recall. An F1 score reaches its best value at 1 and worst at 0. In Table 1 we can find a comparison of performance against the following models, where we used the testing set of the dataset for evaluating each model with the ground truth proposals

Dense Event Captioning

The main task of our model is to detect individual events and describe each with a natural language sentence from input videos. To evaluate the performance of our captioning module we measure the accuracy for each semantic concepts extraction network. We used the Mean Square Error (MSE) metric to evaluate the difference between the generated semantic words and the ground truth words per network, illustrated in Table 2.

We also measure the precision of our captions using Cider metric (Vedantam et al., 2015) that reflects the similarity between two sentences. The average cosine similarity between the candidate sentence and the reference sentences is computed as the Cider score which accounts for both precision and recall.

We average the scores across different tIoU (temporal- intersection-over-union) thresholds of 0.3, 0.5 and 0.7 when captioning the first, second and third events per video in the validation set of ActivityNet Captions “sports and exercising” category.

Although some videos have more than three sentences, we account for results of the first three because at least three sentences are contained in most videos of the dataset. In Table 2, we investigate the performance of static and dynamic semantic networks (D-Sem-N, S-Sem-N), in extracting relevant verbs and nouns with the ground truth, where we used the Mean Square Error (MSE) metric to evaluate the difference between the generated semantic words and the ground truth words per network. Also we evaluated the predicted captions through using the proposals from the event localization module and also by separating the performance of caption generation network from the localization network, by using ground truth temporal locations within test videos. This can be found in Table 3.

Compared Methods

We compare different variants of our model.

- Bi-H: This variant indicates applying our bidirectional proposal method to generate proposals. The hidden states (representing context vectors) from both directions [$\boldsymbol{h}^{\rightarrow}{}_{\boldsymbol{f}}$,$\boldsymbol{h}^{\leftarrow}{}_{\boldsymbol{p}}$] are concatenated and passed to the LSTM decoder of an event.

- Bi-H + Mp: This variant indicates mean pooling current event features and concatenating them with the hidden states (representing context vectors) from both directions [$\boldsymbol{h}^{\rightarrow}{}_{\boldsymbol{f}}$,$\boldsymbol{h}^{\leftarrow}{}_{\boldsymbol{p}}$], for passing them to the decoder of an event. This method uses mean pooling instead of attention to concatenate features.

- Bi-H + TVA: Temporal visual attention (TVA) is used to dynamically fuse visual features along with the local context vectors ($\boldsymbol{h}^{\rightarrow}{}_{\boldsymbol{f}}$,$\boldsymbol{h}^{\leftarrow}{}_{\boldsymbol{p}}$) and passed as visual input within all LSTM gates of the decoder.

- Bi-H + TVA + SemC : This method indicates passing the concatenated attended semantic concepts (SemC) , which is composed of DSC-N and SSC-N, concatenated with the temporal visual attended features (TVA) to an LSTM decoder at each time step within the event proposal.

- Bi-H + TVA + SemC+ CG : This method combining the attended semantic concepts (SemC) , which is composed of DSC-N and SSC-N, with the temporal visual attended features (TVA) using context gating mechanism for balancing them, and passing the result to an LSTM decoder at each time step within the event proposal.

We evaluated the predicted captions through using the proposals from the event localization module and also by separating the performance of caption generation network from the localization network, by using ground truth temporal locations within test videos. This can be found in Table 3.

Network	**Val-accuracy**	**Test-accuracy**
S-Sem-N	80.62	80.72

D-Sem-N	81.44	81.85

Table 2 Semantic concepts feature extraction performance on ActivityNet Captions “sports and exercising” category.

Table 3 Comparing captioning performance of different variants on ActivityNet Captions “sports and exercising” category.

Using GT Proposals		**Using Learnt Proposals**	
Model	**CIDer**	**Model**	**CIDer**
Bi-H	62.2	Bi-H	57.4
MP	62.8	MP	58.1
Bi-H + Mp	64.6	Bi-H + Mp	60.2
Bi-H + TVA	67.5	Bi-H + TDA	63.4
Bi-H + TVA + SemC	71.4	Bi-H + TDA + SemC	67.3
Bi-H + TVA + SemC+ CG	71.8	Bi-H + TVA + SemC+ CG	67.7
[19]	74.2	[19]	60.2

4.3 Experimental Results and Discussion

We experiment our adopted bi-directional LSTM temporal proposal module for detecting event segments that are close to the real segments within the dataset. The F1 score of such module in Table 1 was better than single-direction LSTM which confirms that bidirectional prediction that encodes past, current and future context indeed improves proposal quality, compared to single direction prediction.

The results in Table 3 indicate that, utilizing semantic concepts networks during captioning (Bi-H + TVA + SemC) is more effective than the cases that only used the caption generation without semantics. Also when we used the proposal's hidden states (backward, forward) as "context vectors" and fused them with the event's visual features, via attention mechanism along with the semantic features concepts (Bi-H + TVA + SemC), we got better captioning results, better than using context vectors alone (Bi-H) or event clip features (MP) alone. Applying context gating mechanism for balancing attended visual features and sematic concepts further boosts the performance, which supports that explicitly modeling the relative contribution from event features and contexts in decoding time help better describe the event. Using proposals instead of ground truth events, Table 3 (right), produced a similar trend where adding more context improves captioning.

Based on the results of (Bi-H + TVA), improvements in all scores is noticed when the attention mechanism is applied instead of mean pooling (Bi-H + Mp) for fusing event visual features with the local context vectors. Attending on video visual features at each decoding step allows the model to adequately focus on parts of the event that are relevant to the generated captions with more semantically relevant words. A variant of our framework which is (Bi-H + TVA + SemC) was used for testing the captioning performance on videos with a single event from the MSR-VTT dataset [1] (sports category) test set. Table 4 illustrates the results. We found that it has better caption generation performance than a similar work by [19] which applied temporal attention on mean pooled visual inputs.

Table 4 Performance comparison against other frameworks on MSR-VTT (sports) test set.

Method	CIDer
[19] (2017)	91.1
[5] (2019)	94.4
[6] (2020)	93.7
Ours: Bi-H + TVA + SemC	**97.3**

Our variant outperforms [19, 5, and 6] because we apply attention for the fusion of visual inputs along with context vectors. Also our utilization of semantic features makes sense for better performance.

5 Conclusion

A successful dense video captioning system is considered a fine-grained video understanding task. It differs from early video description methods that produced captions for short video clips that were manually segmented to contain a single event of interest. A dense captioning system must accurately detect the temporal window of each event, and describe these events with a series of coherent sentences. With the recent advances of deep learning, we propose an end to end deep neural network framework that identifies event segments, utilizes context information and extracts semantic concepts, in a single pass, for generating a set of correlated sentences.

The framework is built upon the encoder-decoder structure that continuously encodes the input video stream with three-dimensional convolutional layers and proposes variable-length temporal events that are then transcribed (decoded) into English language words forming sentences. We adopt a bidirectional LSTM framework, for localizing events such that it encodes both past and future contexts where both contexts help localizing the current event better.

We reused the proposal's context information (hidden states) from the localization module as context vectors and dynamically fused those with event clips visual features. Using an attention based mechanism, for the fusion produced superior results compared to using the context alone. Also learning bi-modal semantic features via attending on relevant verbs and nouns that describes the event content effectively, improves the captioning performance and is done while decoding each event in isolation.

References

1. J. Xu, T. Mei, T. Yao, and Y. Rui, "MSR-VTT: A large video description dataset for bridging video and language," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-December, pp. 5288–5296, 2016, doi: 10.1109/CVPR.2016.571.
2. S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, "Translating videos to natural language using deep recurrent neural networks," NAACL HLT 2015 - 2015 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. Proc. Conf., pp. 1494–1504, 2015, doi: 10.3115/v1/n15-1173.
3. L. Yao et al., "Describing videos by exploiting temporal structure," Proc. IEEE Int. Conf. Comput. Vis., vol. 2015 International Conference on Computer Vision, ICCV 2015, pp. 4507–4515, 2015, doi: 10.1109/ICCV.2015.512.
4. M. Zanfir, E. Marinoiu, and C. Sminchisescu, "Grounded Video Captioning," pp. 1–17.
5. C. Yan et al., "STAT: Spatial-Temporal Attention Mechanism for Video Captioning," IEEE Trans. Multimed., vol. 22, no. 1, pp. 229–241, 2020, doi: 10.1109/TMM.2019.2924576.
6. A. Cherian, J. Wang, C. Hori, and T. K. Marks, "Spatio-temporal ranked-attention networks for video captioning," *Proc. - 2020 IEEE Winter Conf. Appl. Comput. Vision, WACV 2020*, pp. 1606–1615, 2020, doi: 10.1109/WACV45572.2020.9093291.
7. Z.Shou, D.Wang, and S.Chang, "Temporal action localization in untrimmed videos via multi-stage CNNs". Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-December, 1049–1058. https://doi.org/10.1109/CVPR.2016.119.
8. Z. F. Tianwei Lin, Xu Zhao*, "TEMPORAL ACTION LOCALIZATION WITH TWO-STREAM SEGMENT-BASED RNN Tianwei Lin , Xu Zhao *, Zhaoxuan Fan Key Laboratory of System Control and Information Processing MOE Department of Automation , Shanghai Jiao Tong University," no. 2, pp. 3–7, 2014.

9. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. C. Niebles, “SST: Single-stream temporal action proposals,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 6373–6382, 2017, doi: 10.1109/CVPR.2017.675.
10. G. Yao, T. Lei, X. Liu, and P. Jiang, “Temporal action detection in untrimmed videos from fine to coarse granularity,” Appl. Sci., vol. 8, no. 10, 2018, doi: 10.3390/app8101924.
11. D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3D convolutional networks,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 International Conference on Computer Vision, ICCV 2015, pp. 4489–4497, 2015, doi: 10.1109/ICCV.2015.510.
12. K. Hara, H. Kataoka, and Y. Satoh, “Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 6546–6555, 2018, doi: 10.1109/CVPR.2018.00685.
13. K. Simonyan, “Two-Stream Convolutional Networks for Action Recognition in Videos arXiv : 1406 . 2199v2 [cs . CV] 12 Nov 2014,” *Proc. 27th Int. Conf. Neural Inf. Process. Syst. - Vol. 1*, pp. 1–11, 2014, [Online]. Available: https://arxiv.org/pdf/1406.2199.pdf.
14. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
15. P. Pan and Z. Xu, “Hierarchical Recurrent Neural Encoder for Video Representation with,” 2015. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-December, 1029–1038. https://doi.org/10.1109/CVPR.2016.117.
16. K. Xu et al., “Show, attend and tell: Neural image caption generation with visual attention,” 32nd Int. Conf. Mach. Learn. ICML 2015, vol. 3, pp. 2048–2057, 2015.
17. H. Wang, A. Kläser, C. Schmid and C. Liu, "Action recognition by dense trajectories," CVPR 2011, 2011, pp. 3169-3176, doi: 10.1109/CVPR.2011.5995407.
18. R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles, “Dense-Captioning Events in Videos,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017-October, pp. 706–715, 2017, doi: 10.1109/ICCV.2017.83.
19. J. Song, L. Gao, Z. Guo, W. Liu, D. Zhang, and H. T. Shen, “Hierarchical LSTM with adjusted temporal attention for video captioning,” IJCAI Int. Jt. Conf. Artif. Intell., vol. 0, pp. 2737–2743, 2017, doi: 10.24963/ijcai.2017/381.