electronics

MDPI

Article

Context Aware Video Caption Generation with Consecutive Differentiable Neural Computer

Jonghong Kim, Inchul Choi and Minho Lee *

School of Electronics Engineering, College of IT Engineering, Kyungpook National University, 80 Daehakro, Bukgu, Daegu 41566, Korea; jonghong89@gmail.com (J.K.); sharpic77@gmail.com (I.C.)

* Correspondence: mholee@gmail.com; Tel.: +82-53-950-6436

Received: 24 June 2020; Accepted: 15 July 2020; Published: 17 July 2020

Abstract: Recent video captioning models aim at describing all events in a long video. However, their event descriptions do not fully exploit the contextual information included in a video because they lack the ability to remember information changes over time. To address this problem, we propose a novel context-aware video captioning model that generates natural language descriptions based on the improved video context understanding. We introduce an external memory, differential neural computer (DNC), to improve video context understanding. DNC naturally learns to use its internal memory for context understanding and also provides contents of its memory as an output for additional connection. By sequentially connecting DNC-based caption models (DNC augmented LSTM) through this memory information, our consecutively connected DNC architecture can understand the context in a video without explicitly searching for event-wise correlation. Our consecutive DNC is sequentially trained with its language model (LSTM) for each video clip to generate context-aware captions with superior quality. In experiments, we demonstrate that our model provides more natural and coherent captions which reflect previous contextual information. Our model also shows superior quantitative performance on video captioning in terms of BLEU (BLEU@4 4.37), METEOR (9.57), and CIDEr-D (28.08).

Keywords: deep neural network; deep learning; context understanding; recurrent neural network; action recognition; memory

1. Introduction

In the last few years, deep learning has significantly contributed to the improvement of visual perception research. This encouraged people in recent vision research to show more interest in challenging problems, such as video understanding. The main goal of video understanding is to describe the contents of a video in natural language automatically. Compared to image captioning, which describes a still image, video understanding is a more challenging task as information in a video is far more complicated. We need to capture not only the spatial contents of video (objects, scenes), but also the temporal dynamics (actions, context, flow) within the video sequences for adequate video description. Recent advances in 3D convolutional neural networks (CNNs) [1] provide a method to yield semantic representation of each short video segment which also embeds temporal dynamics. However, for longer and complicated video sequences (such as 120 s long in the ActivityNet dataset [2]), the context understanding of a video becomes more important for generating natural descriptions. Since long videos involve multiple events ranging across multiple time scales, capturing diverse temporal context between events is the key for natural video description with context understanding. In the following, we introduce traditional video captioning models and their limitations for video context understanding. We then propose our approach to this problem.

The conventional deep learning based video captioning models adopt gated recurrent neural networks (RNN), such as long short-term memory (LSTM) or gated recurrent unit (GRU), with the encoder-decoder architecture [3–5]. Those models encode each input video sequence into semantic representations and send it to a decoder to generate video captions. However, those approaches have following limitations:

1. They cannot generate natural captions for long videos with diverse and complex events.
2. They suffer from a lack of context understanding.

The conventional RNN is not sufficient to encode information with long-term dependency in long video sequences. Although LSTM or GRU partially address such an issue, long-term dependency is still an unsolved problem for long sequential data. Additionally, those conventional captioning models hardly maintain the contextual information included in a video sequence. Due to the memory limitation in the RNN models, the conventional video captioning models only work for short video clips with simple scenes and are not applicable to long videos consisting of multiple complicated events.

To overcome this limitation, in this paper, we propose a novel context aware video captioning model which can generate captions based on temporal contextual information in a long video as shown in Figure 1. To focus on the temporal context alone, we divide a long input video into event-wise sub-video clips and leverage external memory to understand temporal contextual information in the video. To reliably store and retrieve temporal contextual information, we adopt a differentiable neural computer (DNC) [6]. DNC naturally learns to use its internal memory for context understanding in a supervised fashion and it also provides the contents of its memory as an output. In our proposed model, we consecutively connect the DNCs based captioning models (DNC augmented LSTM) with this memory information which reflects the context, and sequentially train each language model to generate captions for each sub scenes. In our experiments, we show that the proposed model generates temporally coherent sentences by using previous contextual information, and compare the captioning performance with other state-of-the art video captioning models. Additionally, we show the superior performance of our model based on quantitative measures, such as Bilingual Evaluation Understudy (BLEU) [7], Metric for Evaluation of Translation with Explicit Ordering (METEOR) score [8], and Consensus-based Image Description Evaluation (CIDEr-D) [9].

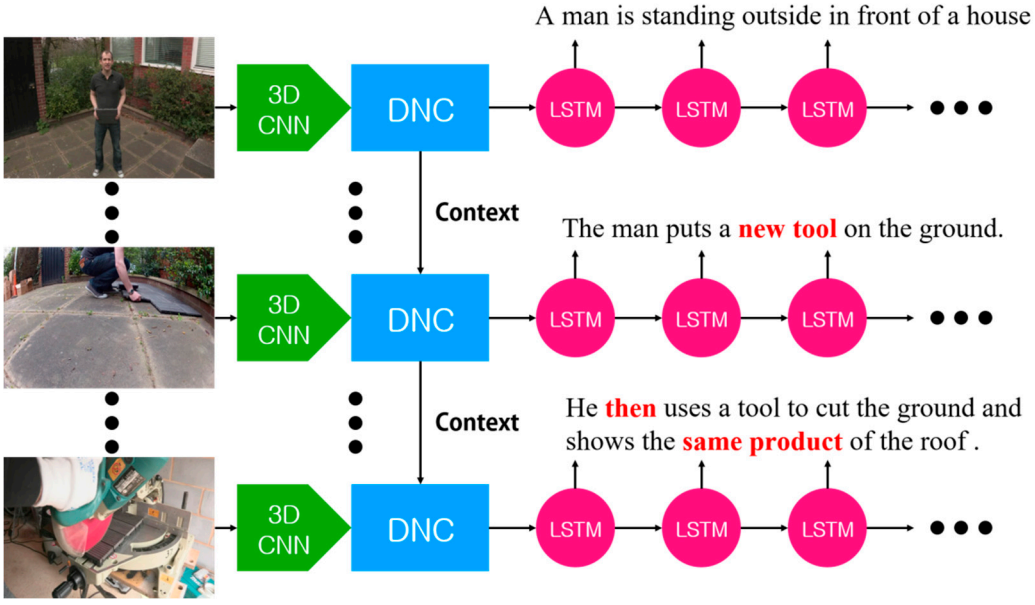


Figure 1. Context aware video caption generation is required to describe situationally complicated video in more precise and natural manner. We apply a consecutively-connected DNC architecture to understand the context in the video.

Context Aware Video Caption Generation

The context aware video caption generation regards how much the generated output is relevant to its context. In Figure 1, the generated sentence includes 'new tool' which indicates our proposed model understands that the 'tool' of current input scene is 'new' one. Additionally, another generated sentence includes 'then', which indicates the proposed model understands causality of events. Such abilities only can be accomplished by context aware caption generation.

From the current scene, 3D CNN extracts valuable information as feature map. Based on the extracted feature map, the DNC memorizes current scene information and passes through its current state to the next DNC. The second DNC uses not only current input information but also initialized by previous DNC state information. Finally, the second DNC comprehends contextual relationships between previous states and the current input by understanding the current input based on previous states. Such a process accumulates consecutively, therefore, the final DNC can accommodate context of the events.

2. Related Works

The early video captioning studies focused on extracting semantic content, such as subject, verb, or object, and associate them with visual elements in the scenes [10–12]. For instance, there is a study [11] that constructs a factor graph model to obtain the confidence of semantic contents and finds the optimal combination of them for sentence template matching. However, such an early model only works for specific videos and the number of possible expressions is limited. In contrast, recent research [13,14] shows that the deep learning based approach is effective for video-based language modeling tasks when it is trained with large dataset including vast amounts of linguistic information.

The earlier studies on deep learning-based video captioning use mean pooling on the feature map from the pre-trained convolutional neural network (CNN) to obtain feature representations of every input video frame and apply RNN for language modeling [14]. However, this method is limited only for video clips with short and static backgrounds. With the success of neural machine translation (NMT), the LSTM-based encoder-decoder structure, which is known as the sequence-to-sequence model, is also applied for video captioning [5]. They obtain semantic representations of video frames from the pre-trained CNN and provide them as input to the LSTM encoder to obtain the final hidden states. Then they optimize the loss function of the LSTM decoder for one-step ahead prediction to generate subsequent words. However, since the sentence generation of a decoder only depends on the output of an encoder, they cannot obtain good performance for long videos. In order to address this problem, the attention mechanism [15] is introduced to video captioning [16]. Through the attention mechanism, RNN can generate each word based on soft attention over the temporal segments of a video. There is a study that adopts such soft attention and visualizes the activated attention region when it generates a word for image captioning [17]. They apply the CNN feature map vector a_i and the LSTM hidden state h_{t-1} to the attention model [15] and obtain the attention weight $\alpha_{t,i}$ which indicates a relation between them. Then they train the LSTM decoder with the weighted sum of CNN feature map as an initial state to implement the soft attention.

For longer and coherent captioning, researchers also try to consider context information [13,18]. They introduce a hierarchical RNN to encode both local and global contexts of a video. In their model, the first level of hierarchy learns local temporal structure of each subsequence and the second level of hierarchy learns global temporal structure between subsequences [18]. They also applied attention to each layer of the hierarchy to obtain a richer representation for the video captioning which successfully increased the performance score of METEOR and BLEU. There is another type of hierarchical structure which stacks RNN for considering contextual information over the RNN for sentence generation [13]. In that model, the higher-level RNN combines its contextual hidden states with the embedded sentence generated from the lower-level RNN to decide an initial state for the generation of the next sentence.

The closest work to our model is the dense video captioning which generates captions for every event in the videos [19]. They adopt Deep Action Proposals (DAPs) [20] in order to estimate the start and

end time of each event in the video and train DAP with the video captioning model. To reflect the past and future event contexts to the current video caption generation, they apply the attention mechanism to the hidden states of the LSTM which encodes each sub video clip to show its improvement in overall context understanding. There is a recent study that introduces descriptiveness-driven temporal attention which is an improved version of the temporal attention [21]. They applied holistic attention score which represents descriptiveness of each clip composing a video to increase the attention weights of descriptive clips.

Previous works applied attention mechanism and hierarchical structure to include the contextual information in video captioning. However, the context awareness requires to memorize both the sequence of important events and the relationship between them.

In this paper, we propose a new method that can overcome such limitations of previous video captioning models by leveraging external memory (DNC) for context understanding [6]. Our model not only can generate natural captions for each event in a video but also reflects context information between related events for coherent captioning. There is a recent study that introduces descriptiveness-driven temporal attention which is an improved version of the temporal attention [21]. They applied holistic attention score which represents descriptiveness of each clip composing a video to increase the attention weights of descriptive clips.

Previous works applied attention mechanism and hierarchical structure to include the contextual information in video captioning. However, the context awareness requires to memorize both the sequence of important events and the relationship between them.

In this paper, we propose a new method that can overcome such limitations of previous video captioning models by leveraging external memory (DNC) for context understanding. Our model not only can generate natural captions for each event in a video but also reflects context information between related events for coherent captioning.

3. Video Captioning with DNC

3.1. Differentiable Neural Computer (DNC)

Differentiable neural computer (DNC) started from a Turing machine. Turing machines are abstract modern computer structures which show that all computations are possible given the appropriate external memory and algorithms [4]. Google Deep Mind proposed the Neural Turing Machine (NTM), a system that combines neural networks and external memory to implement a differentiable Turing machine, and in 2016, a Nature paper proposed an improved version of the NTM model, DNC. In their paper, they demonstrated that DNC can effectively learn how to use memory to deal with complex and structured data such as Q&A (bAbI), family tree, and London's subway maps [6]. DNC consists of a controller and a memory, and the controller transmits a control signal to the memory unit in an interface vector. The interface vector contains several parameters related to memory operation, and each parameter determines the weighting factor, which is the degree involved in reading or writing memory. In the process of finding the correct answer through learning, the controller is trained to output an interface vector that gives the optimal weighting factor. In other words, the controller learns how to determine the weighting factor that determines where, in what order, and how much information is read or written in memory, all of which are determined by the interface vector. Each component included in interface vector is shown in Table 1.

DNC can perform content-based addressing to find useful information by calculating the similarity between the content of a memory and a key vector, and location-based addressing to search for information in the order entered in memory or in reverse order, regardless of similarity. This is why DNC is able to respond flexibly in understanding the complex nature of data:

$$c_t = C(M, k, \beta)[i] = \frac{exp(D(k, M[i, \cdot])\beta)}{\sum_j exp(D(k, M[j, \cdot])\beta)} \quad (1)$$

$$D(u,\ v) = \frac{u \cdot v}{|u||v|} \tag{2}$$

where $c_t \in [0, 1]$ means content-based weighting, and is determined by the cosine similarity between the key vector belonging to the interface vector and the information vector in memory. DNC allows content-based weighting to flexibly determine from the data how much to read or write to the information in memory:

$$w_t^{r,i} = \pi_t^i[1]b_t^i + \pi_t^i[2]c_t^{r,i} + \pi_t^i[3]f_t^i \tag{3}$$

$$r_t^i = M_t^T w_t^{r,i} \tag{4}$$

Equation (3) calculates read weighting $w_t^{r,i} \in [0, 1]$, which determines how much information in memory is read, and Equation (4) calculates read vector r_t^i, which means information read from memory through context-based addressing. $\pi_t^i[1]$, $\pi_t^i[2]$, and $\pi_t^i[3]$ mean three read modes: backward, content-based, and forward, and read mode is assigned to each read head. Read weighting $w_t^{r,i}$ is defined as a weighted sum of each read mode vector, backward weighting b_t^i, content-based weighting $c_t^{r,i}$, and forward weighting f_t^i. Finally, R read vectors r_t^i are obtained through the matrix product of memory matrix M_t and read weighting $w_t^{r,i}$:

$$w_t^w = g_t^w\left(g_t^a a_t + \left(1 - g_t^a\right)c_t^w\right) \tag{5}$$

$$M_t = M_{t-1}\left(E - w_t^w e_t^T\right) + w_t^w v_t^T \tag{6}$$

Equation (5) shows write weighting $w_t^w \in [0, 1]$ to determine how much information to allocate to memory, and Equation (6) shows the update process of memory matrix M_t a_t is allocation weighting, which introduces a usage vector, which is a value related to the frequency of memory usage, so that the usage vector has a small value, that is, a large value at a memory address that has not been used. The allocation gate $g_t^a \in [0, 1]$ in Equation (5) is learned to have a large value when memory allocation occurs. It enables flexible memory allocation by determining the superiority of location-based addressing and content-based addressing. Finally, the memory erases the information from memory by subtracting the multiplication of erase vector e_t and write weighting w_t^w from the previous memory matrix M_{t-1}, and the information is allocated to memory by adding the multiplication of write vector v_t and write weighting w_t^w as shown in Equation (6).

Table 1. Components of interface vector.

Notation	Name	Notation	Name
$k_t^{r,i}$	Read keys	v_t	Write vector
$\beta_t^{r,i}$	Read strength	f_t^i	Free gates
k_t^w	Write key	g_t^a	Allocation gate
β_t^w	Write strength	g_t^w	Write gate
e_t	Erase vector	π_t^i	Read modes

3.2. A Single DNC-LSTM-Based Video Caption Model

Our proposed model sequentially connects and trains a single video caption generation model with DNC for temporal context awareness. As shown in Figure 2, a single video caption generation model is an encoder-decoder network integrated with DNC memory. It encodes a video clip with pre-trained 3D-CNN and generates sentences from the information stored in DNC memory by using an LSTM decoder.

We use a 4096-dimensional feature map obtained from the pre-trained 3D-CNN [1] to extract spatial-temporal features of a video. For this process, we divide video sequences into a number of small video clips with each 16 frames *(*$\delta = 16$*)* and extract feature vectors for each clip. For more

temporally coherent feature extraction, we apply this method on every video clip by overlapping eight frames as follows:

$$features_t = \Phi\left(frames_{\delta(t-0.5):\delta(t+0.5)}\right) \tag{7}$$

where $features_t$ are the extracted features at time step t, $frames_{\delta(t-0.5):\delta(t+0.5)}$ are the selected frames in the current video clip, and $\Phi(\cdot)$ is the 3D CNN which generates a 4096-dimensional feature map from the last fully-connected layer.

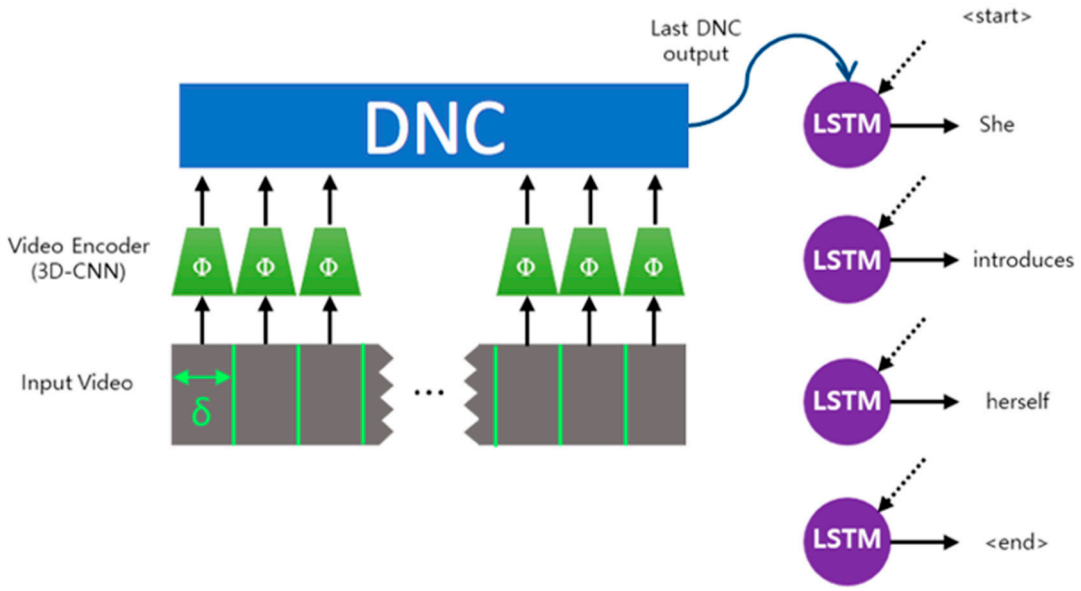


Figure 2. Basic structure of a single video caption generation model. The encoder part of the encoder-decoder structure is replaced with DNC. The features extracted from 3D CNN are fed into the DNC for each time step and stored in the DNC memory. After encoding all input, the final DNC output is used as the initial state of the LSTM decoder for caption generation.

After this process, to reflect the previous information in the memory of the DNC to current information, we concatenate the read vector, which reflects the contents of past DNC memory with the current input feature vector and provide them as input to the DNC controller. Since the input to the DNC controller includes the sequence of 3D CNN features, input x_t on $(time\ step) = t$ is as follows:

$$x_t = concat(r_{t-1}, features_t),\ t \in [0, T] \tag{8}$$

where r_t is the read vector at the time step t. The DNC output of last time step T, which is the concatenation of read vector r_T and controller output vector v_T, is projected on the output space. This value is used as the initial state s_0^{dec} of LSTM decoder as in Equation (9). The decoder is trained to generate sentences based on the value of s_0^{dec}:

$$s_0^{dec} = W_{out}^{DNC} concat(r_T, v_T) \tag{9}$$

where $W_{out}^{DNC} \in \mathbb{R}^{d_h \times (d_{read} + d_{controller_{out}})}$, d_h is the number of LSTM decoder hidden unit, d_{read} is the size of read vector and $d_{controller_{out}}$ is the size of LSTM controller output vector. The decoder has the LSTM structure which takes input of embedded words as shown in Equation (10). In Equation (11), the output of LSTM at each time step h_t^{dec} predicts the one-hot vector of a next word by applying the Softmax function to the outcome of the fully connected output layer with the nodes of vocabulary size as:

$$\left(c_t^{dec}, h_t^{dec}\right) = s_t^{dec} = LSTM^{dec}\left(s_{t-1}^{dec}, W_{emb}^{dec} w_t^{in}\right) \tag{10}$$

$$pred_t^{dec} = softmax\left(W_{pred}^{dec} h_t^{dec}\right) \tag{11}$$

where $W_{emb}^{dec} \in \mathbb{R}^{d_{emb} \times Vocab}$, $W_{pred}^{dec} \in \mathbb{R}^{d_h \times Vocab}$ and c_t^{dec} is the LSTM decoder cell state where $Vocab$ is the size of the vocabulary and d_{emb} is the size of the word embedding vector of the decoder. The w_t^{in} is the one-hot vector with the size of vocabulary and is fed as an input to each decoder step. We can obtain the vector by shifting the target sentence by one step for the one-step ahead prediction. The W_{emb}^{dec} is the word embedding matrix and the W_{pred}^{dec} is a matrix for projection of LSTM output to a vocabulary size vector.

The loss function is defined by the cross-entropy between the LSTM output vector and the one-hot vector of a target word, and optimized by the Back-Propagation Through Time (BPTT) algorithm as follows:

$$loss^{dec} = -\frac{1}{T}\frac{1}{N}\frac{1}{Vocab} \times \sum_t \left(\sum_i onehot\left(w_t^{target}\right) \times log\left(pred_t^{dec}\right) \right) \tag{12}$$

where $i \in [0, Vocab]$, $t \in [0, L]$, and N is the size of the mini-batch. Since composing long video samples with several mini-batches can cause excessive zero-padding, we set $N = 1$ in our proposed model. The w_t^{target} is t^{th} word of the target sentence which is converted to the one-hot vector of vocabulary size. The DNC learns optimal memory operations to generate a target sentence and the LSTM decoder learns to generate sentences when the video scene representation is given.

Through the association operation which is based on the similarity between the given data and stored information in a memory, the DNC can retrieve information stored in the memory. In our proposed model, we take advantage of such characteristic of the DNC for understanding the context of complex and long videos.

3.3. Consecutive DNC-Based Video Caption Model

In this paper, we have two hypotheses:

1. The information stored in external DNC memory has its own unique context information.
2. The temporal contextual information can be obtained through the connections of (1) over time.

Based on the above hypotheses, the main idea of our model is to provide a temporal context to the video caption generation by passing various components involved in DNC memory operations and its mechanism to the next stage of DNC.

In other words, we can utilize abstracted information of accumulated input data in DNC memory and the values involved in the memory I/O operation as a medium for context awareness. Under this assumption, we define the context vector $Context_k$ of k^{th} sub-scene with an input sequence length T as follows:

$$Context_k = \left(M_T^k, u_T^k, p_T^k, L_T^k, w_T^{w,k}, w_T^{r,k}, r_T^k\right) \tag{13}$$

where $k \in [0, K]$. The meaning of each component of the $Context_k$ is follows:

- Memory matrix M_T^k: Abstractive representation of the input data updated by content-based addressing and location-based addressing.
- Usage vector u_T^k: Frequency of usage. The more frequently the memory is used, the larger the value.
- Precedence vector p_T^k: Memory allocation priority. When a memory block is assigned with data, it decreases. It differentiates memory allocation priority to reduce interference between the memory blocks.
- Link matrix L_T^k: The order of each memory block usage. $L_T^k[i, j]$ becomes larger if the j^{th} memory block, right after the i^{th} memory block is more frequently used.
- Write weight $w_T^{w,k}$: The degree of a given input data is reflected to memory. The larger the value, the more information is stored in memory.

- Read weight $w_T^{r,k}$: The degree of information read from memory. The larger the value, the more information is read from memory and reflected in an output.
- Read vector: Information read from a memory through three types of read modes—content-based, forward, and backward.

As shown in Figure 3, each DNC memory for each sub-video clip is initialized with the context vector $Context_k$ generated from its previous DNC for context awareness. Based on this structure, we sequentially train each LSTM decoder for coherent caption generation. If we represent the model in Section 3.1 as $G(\cdot)$, the $(k-1)^{th}$ context vector as $Context_{k-1}$, feature sequence obtained from the 3D-CNN for k^{th} sub-scene as $features^{k}_{0,\cdots,(T-1)}$ and the result of sentence generation as $pred^{dec,k}_{0,\cdots,(L-1)}$, then, our entire video captioning model can be described as Equation (14).

$$pred^{dec,k}_{0,\cdots,(L-1)} = G\Big(features^{k}_{0,\cdots,(T-1)}, Context_{k-1}; \theta\Big) \tag{14}$$

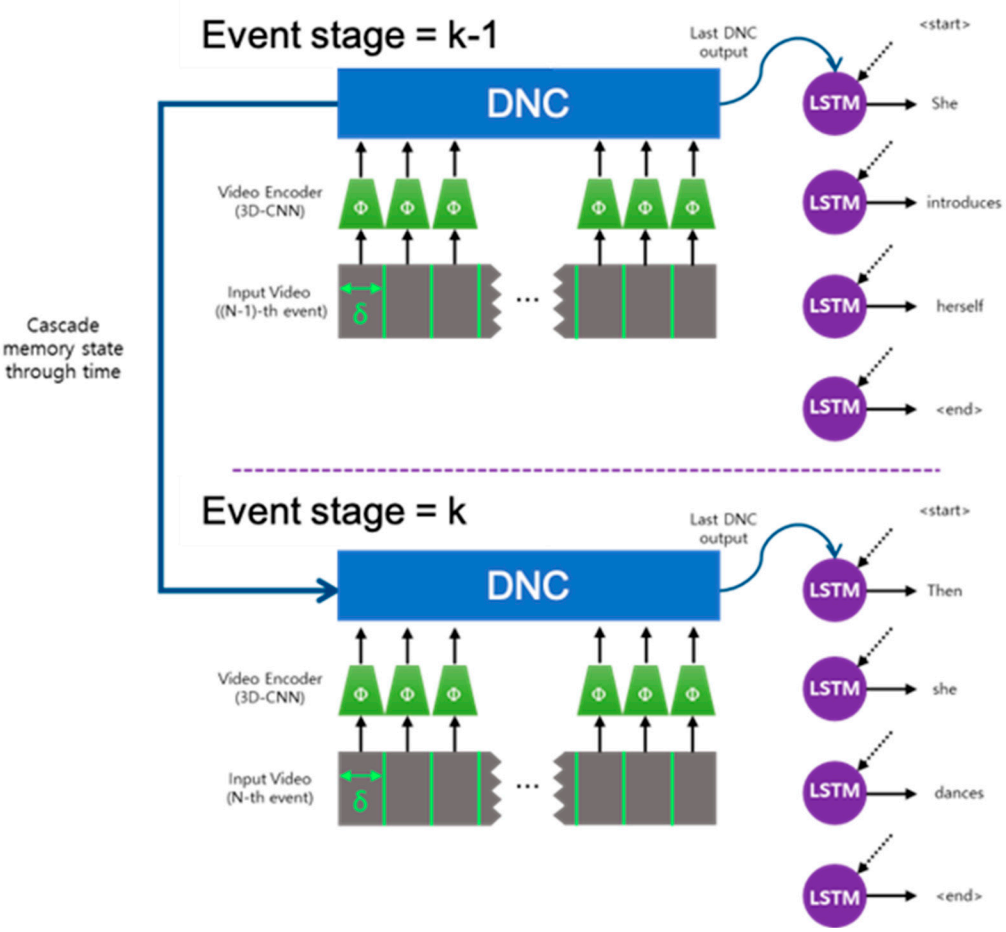


Figure 3. Consecutive DNC based structure to understand context and generate video captions. Each event stage is identical to the single DNC model. However, the state of the k^{th} DNC is initialized with the final state of the $(k-1)^{th}$ DNC. Therefore, the consistency in context can be preserved.

Based on the BPTT algorithm, our model is trained by sequentially optimizing the cross-entropy loss function between the sentence $pred^{dec,k}_{0,\cdots,(L-1)}$ from each sub-scene and a target sentence $w^{target,k}_{0,\cdots,(L-1)}$

over all sub-scenes as shown in Equation (15). The DNC and the LSTM share their model parameters over every stage k for context understanding and sentence generation:

$$loss_{context}^{dec,k} = -\frac{1}{T}\frac{1}{N}\frac{1}{Vocab} \times \sum_{t}\left(\sum_{i} onehot\left(w_t^{target,k}\right) \times log\left(pred_t^{dec,k}\right)\right) \quad (15)$$

where $i \in [0, Vocab]$ and $t \in [0, L]$. The proposed model learns how to read or write various connection patterns between each sub-scene to the external memory and flexibly utilize the context information for video captioning. In addition, by reading the context information accumulated in the memory through a DNC read vector, and provide it as the initial state of the LSTM decoder for training, it is possible to generate the sentences with context understanding.

4. Experimental Results

For the performance evaluation, we compare our model with other state-of-the-art video captioning approaches [14,19,21–23] with respect to the context awareness. We have performed four experiments with the ActivityNet Caption dataset [19]. First is learning the curve comparison, which indicates that the proposed model is computationally efficient, as shown in Figure 4. Second is the 'without context' experiment, which indicates the efficiency of DNC itself. Third is the 'with context' experiment, which indicates how our proposed consecutive DNC outperforms other approaches. Finally, the last experiment is a qualitative evaluation for generated caption examples, which indicates that our results include more relevant contextual information. Since the goal of our approach is understanding the context of the video without explicitly searching for event-wise correlation, dense video captioning models are not appropriate for comparison. Therefore, we select only video captioning models which considers context from video for our experiment. We also show the effect of the DNC memory connections over time in our model. In our experiments, we focus on the context awareness of generated captions. To evaluate the model performance independent of sub-scene localization, we assume that event localization is already performed. The 3D CNN feature sequences are extracted based on the specified start and end time of each sub-scene in the ground truth dataset, and used for training and testing of our model.

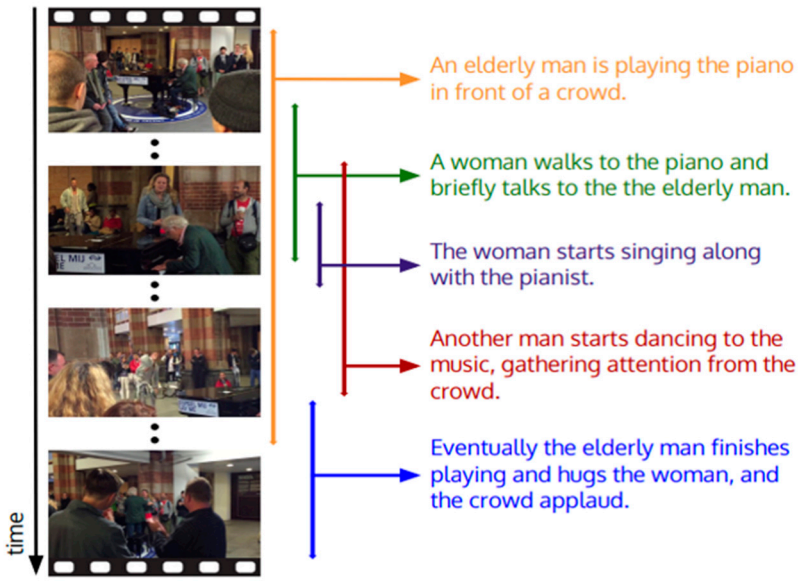


Figure 4. An example of the ActivityNet caption dataset. Each caption description is related with not only the current scene but also past situations. Therefore, the ActivityNet dataset is suitable to show the context awareness performance of our proposed model.

The dataset used in our experiments is ActivityNet Caption dataset [19] which is based on ActivityNet version 1.3 [24] and consists of about 20,000 YouTube videos. Each video has an average length of 180 s and each datum sample includes the captions which are composed of start/end time and description of the sub-scenes. Each sample includes three sub-scenes in average and each caption is a sentence consisting of average 13.5 words. The captions are prepared while considering the causal relationships between events. The number of training/validation/test videos are 10,024/4926/5044, respectively, and the total number of sentences is 100,000.

4.1. Model Training

To extract the spatio-temporal features from a video, we apply a pre-trained 3D CNN which is trained on sports 1M dataset [1]. For this process, we define 16 video frames as one clip and input them to the 3D CNN to extract a 4096-dimension feature vector of the last fully-connected output layer. In order to extract detailed features, slide-windowing is performed on the video samples by overlapping eight frames at a time. For sentence preprocessing, the PTB tokenizer included in the Stanford CoreNLP tool [25] is used. The word dictionary for converting words to integers is constructed based on the sentences contained in the training and validation datasets. Each word is converted to a one-hot vector, and then converted to a dense vector expression by multiplying the word with a matrix for word embedding, and is used as input to the decoder.

The DNC controller uses LSTM with 256 hidden units. In addition, the number of DNC memory blocks is 256, the size of the vector stored in a memory is 64 dimensions, and the read head for the read vector is four, in total. The read vector of the DNC is projected to a vector with a size of 1024 to obtain a final output, and this vector is used as the initial state of the LSTM decoder. The number of hidden units of the LSTM decoder is set to 1024, and each word is converted into a 300-dimension embedding vector. To avoid over-fitting, we apply the dropout [26] with 0.3 ratio on every I/O layer of the LSTM layers.

For training, the ADAM optimization algorithm is used, and the learning rate is set to 2×10^{-4} and the momentum decay parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ are used according to the method proposed in [27]. As shown in Equation (15), the cross-entropy between the output of the LSTM decoder and the one-hot vector of a target word is used for the loss function. Since configuring video consisting of multiple sub-scenes into multiple mini-batches can results in excessive zero-padding, we set the size of mini-batch as 1.

After training the k^{th} scene, to provide the context information accumulated in the DNC to the $(k+1)^{th}$ scene, we construct a $Context_k$ tuple as described in Equation (13) and set it as the initial value of the DNC memory for the training of the $(k+1)^{th}$ scene. The same procedure is performed sequentially for all following sub-scenes. We measure the loss and 4-g BLEU score (B@4) of the training data for every 100th time point and check the progress by measuring the loss and BLEU score of the validation data for every 1000 time point. After six epochs of training, when the loss and BLEU scores are converged for validation data, we finish the training process. In our experiment, the entire training takes 12 h with the computing power of NVIDIA QUADRO GV100 of 5120 CUDA cores, 640 tensor cores and 32GB GPU memory.

For the evaluation of a single DNC based captioning model, it is not connected with any other DNC memories over time and the $Context_k$ tuple is also not used. The initial value of the single DNC model is always set to default so that the caption can be generated only from the current input video sequences without any consideration for the context between events. All other conditions are same, and the training is continued for six epochs and finished. Figure 5 shows the learning curve of our model conditioned on 'with context' and 'without context'. Both learning curves which are almost similar indicate that even though we additionally include our proposed model, it does not increase the problem complexity. Each line with the color of blue and brown represents the BLEU score of training and validation, respectively, and each green and orange correspond to the loss of training and validation, respectively.

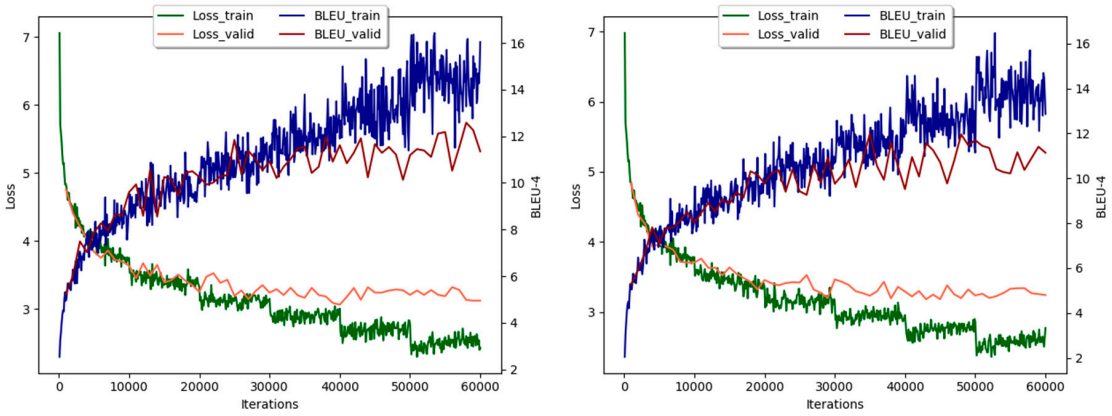


Figure 5. Learning curve of 'with context' (**left**) and 'without context' (**right**). Each line with the color of blue and brown represents the BLEU score of training and validation, respectively, and each green and orange line corresponds to the loss of training and validation, respectively.

4.2. Performance Comparison with Other Approaches

Tables 2 and 3 show the results of the quantitative performance comparison between the proposed model and other models in the video captioning field with the ActivityNet Caption dataset. For the performance measurements, BLEU [7], METEOR [8], and CIDEr-D [9] are used.

In the 'without context' comparison, [14,22] trained each sub-scene and its caption as a single sample, and the context between the scenes is disconnected as in the case of 'without context' condition in our model. In LSTM-YT [14], feature maps from the pre-trained VGG network are extracted and the result of mean pooling over the time axis is used as the initial state of the LSTM decoder. S2VT [22] has an encoder-decoder structure in which the mean pooling is replaced with an LSTM encoder. Those two models are well-known approaches which use CNN-extracted features for caption generation, but not considering contextual information. Those models only consider temporal sequential information in their structures. Therefore, those approaches are suitable for 'without context' comparison to show the caption generation performance of a single DNC model. H-RNN [23] uses two RNNs, one for a sentence generator, and the other for determining an initial state of a generator for the next sub-scene description. TempoAttn [19] presented the dataset used in this experiment, and we adopt their model as a baseline of comparison for our study. In order to reflect the past and future contexts to the current sentence generation [19], applied an attention mechanism to the hidden states of the LSTM which encodes each scene for context understanding. DVC [21] is the most recent study to have applied a holistic attention score on an attention mechanism to distill descriptive video clips. Those three selected models are known as context aware approaches. Those models consider specific structures to address context information. Therefore, we can show how much our proposed model generates contextually more relevant results in quantitative and qualitative evaluation compared to those models. The evaluation is performed for [14,22] with the condition of 'without context' and [19,21,23] 'with context', respectively.

Table 2. Comparison with other video caption generation models not considering context for the Activity Captions validation set with ground truth proposals. Bold numbers indicate the best results compare to other approaches.

	BLEU@1	BLEU@2	BLEU@3	BLEU@4	METEOR	CIDEr-D
LSTM-YT [14]	18.40	8.76	3.99	1.53	8.66	**24.07**
S2VT [22]	18.25	8.68	4.02	1.57	**8.74**	24.05
Without context (ours)	**21.4**	**10.3**	**5.72**	**3.51**	8.56	23.87

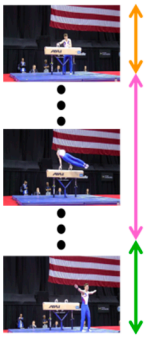
As shown in Table 2, compared to [14,22] with LSTM, the proposed DNC-based 'without context' model produces an overall higher score for BLEU and comparable scores for METEOR and CIDEr-D.

In this result, we can see that using DNC as a video encoder instead of LSTM can improve the video captioning performance in terms of BLEU.

Table 3. Comparison with other video caption generation models considering context for Activity Captions validation set with ground truth proposals. Bold numbers indicate the best results compare to other approaches.

	BLEU@1	BLEU@2	BLEU@3	BLEU@4	METEOR	CIDEr-D
H-RNN [23]	18.41	8.80	4.08	1.59	8.81	24.17
TempoAttn [19]	18.13	8.43	4.09	1.60	8.88	25.12
DVC [21]	19.57	9.90	4.55	1.62	**10.33**	25.24
With context (ours)	**23.4**	**11.9**	**6.88**	**4.37**	9.57	**28.08**

The comparison results in Table 3 show that our proposed model outperforms all other models in terms of BLEU and CIDEr-D score considerably. Additionally, our model's METEOR score is superior to the baseline Temporal Attention model [19] and the H-RNN model [23], but slightly lower than DVC [21]. Moreover, compared to the proposed 'without context' model result in Table 2, the proposed 'with context' model shows significant improvement in performance. The quantitative analysis results indicate the excellence of the proposed consecutive DNC structure for context awareness. Since the goal of our model is not dense captioning, but context-aware captioning, quantitative analysis is not sufficient to measure how well the context is reflected in the generated descriptions. Therefore, we will show a more detailed result of our proposed model through qualitative analysis. In Figure 6, we compare the generated captions of our 'with context' model, 'without context' model and TempoAttn [19]. Figure 6a shows that TempAttn repeats the same description for the second scene, although the scene has already changed. However, for the same case, our model generates the sentences with words of contextual meaning, such as the endings and beginnings, which are marked in bold, implies that our description contains more context-related words than other models. In the case of the 'without context' model, even though the gymnast's performance is already finished in the last scene, it cannot recognize the situation and generates an incorrect description, "beginning performing". In contrast, the 'with context model' generates more natural and coherent sentences compared to all other models by using the words with contextual meaning, such as "then", "the new tool" (different from a previous tool), "the same product" (same as a previous one), etc.



GT	*TempoAttn*	*Ours(without context)*	*Ours(with context)*
A male gymnast is on a mat in front of judges preparing to begin his routine .	He mounts the beam then does several flips and tricks .	A gymnast is standing on a stage .	A gymnast mounts a beam in a gym .
The boy then jumps on the beam grabbing the bars and doing several spins across the balance beam .	He does a gymnastics routine on the balance beam .	He is doing gymnastics on the parallel bars .	**After** he mounts the beam and **begins** performing several tricks and tricks .
He then moves into a hand stand and jumps off the bar into the floor .	He does a gymnastics routine on the balance beam .	A gymnast is seen standing ready to a set of uneven bars and begins performing a routine on the parallel bars .	He dismounts , raising his arms up and lands on the mat .

(**a**)

Figure 6. *Cont.*

GT	*TempoAttn*	*Ours(without context)*	*Ours(with context)*
A man is standing outside holding a black tile .	A man is seen speaking to the camera while holding up a bucket and begins putting the wall .	A man is standing in front of a house .	A man is standing outside in front of a house .
He starts putting the tile down on the ground	A man is seen needing down on a roof and begins using a tool on the carpet .	A man is putting a wooden board on the floor .	The man puts a **new tool** on the ground .
He cuts the tile with a red saw .	A man is seen speaking to the camera and leads into him holding knives and sharpening a board .	A man is using a vacuum to clean the roof of a roof .	He **then** uses a tool to cut the ground and shows the **same product** of the roof .
He sets chairs and flowers on the tile .	The person then walks around the table painting the fence .	We see the opening title screen .	The screen fades to black .

(**b**)

Figure 6. Comparison between generated captions. Our proposed model, consecutive DNC ('with context'), and without consecutive connection of DNC ('without context') are compared to TempoAttn. 'GT' represents ground truth: (**a**) is a gymnast example and (**b**) is a tile work example.

According to the qualitative analysis, we can easily understand that the frequent usage of pronouns or conjunctions for contextual expressions can be found compare to other approaches. In order to come out of various concatenations, the connection must be natural considering the context, so we can realize that the context is well considered even though it is disadvantageous in some cases of numerical evaluations. Furthermore, the performance of the consecutive DNC captioning model with contextual connection is superior to the single DNC captioning model without context consideration. This result demonstrates the effectiveness of the contextual connection of the DNC-based caption generation model in learning temporal context.

5. Conclusions

In this paper, we propose a new video captioning model that comprehends video with context information and generates natural and coherent captions. We showed the superior captioning performance of our model for the video with context information when compared to other state-of-the-art video captioning models. Our model leverages the external memory (DNC) for the context information management in a single video captioning model and sequentially connects several single captioning models to enhance temporal context understanding in video. This linking of external memory (with context) via contextual information vector showed significant improvement in video context understanding [14,19,21–23]. From those results, we can conclude that the introduction of DNC memory for managing context information improves not only the performance of video captioning but also the understanding of contexts in a video. In our future work, we will improve the feature extraction method and include event localization method to our proposed model. Additionally, we will combine Transformer [28] and BERT [29] with our proposed DNC based model for better performance.

Author Contributions: Conceptualization, J.K. and M.L.; methodology, J.K.; software, J.K.; validation, J.K., I.C. and M.L.; formal analysis, J.K.; investigation, J.K.; writing—original draft preparation, J.K.; writing—review and editing, J.K., I.C. and M.L.; visualization, J.K.; supervision, M.L.; project administration, J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea government (MSIT) (2016-0-00564, Development of Intelligent Interaction Technology Based on Context Awareness and Human Intention Understanding).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 221–231. [CrossRef] [PubMed]
2. Heilbron, F.C.; Niebles, J.C. Collecting and annotating human activities in web videos. In Proceedings of the International Conference on Multimedia Retrieval, Glasgow, UK, 1–4 April 2014; p. 377.
3. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
4. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
5. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 3104–3112.
6. Graves, A.; Wayne, G.; Reynolds, M.; Harley, T.; Danihelka, I.; Grabska-Barwińska, A.; Colmenarejo, S.G.; Grefenstette, E.; Ramalho, T.; Agapiou, J. Hybrid computing using a neural network with dynamic external memory. *Nature* **2016**, *538*, 471. [CrossRef] [PubMed]
7. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.-J. BLEU: A method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Philadelphia, PA, USA, 7–12 July 2002; pp. 311–318.
8. Banerjee, S.; Lavie, A. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Ann Arbor, MI, USA, 25–30 June 2005; pp. 65–72.
9. Vedantam, R.; Lawrence Zitnick, C.; Parikh, D. Cider: Consensus-based image description evaluation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4566–4575.
10. Krishnamoorthy, N.; Malkarnenkar, G.; Mooney, R.; Saenko, K.; Guadarrama, S. Generating natural-language video descriptions using text-mined knowledge. In Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, Bellevue, WA, USA, 14–18 July 2013.
11. Thomason, J.; Venugopalan, S.; Guadarrama, S.; Saenko, K.; Mooney, R. Integrating language and vision to generate natural language descriptions of videos in the wild. In Proceedings of the COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, 23–29 August 2014; pp. 1218–1227.
12. Xu, R.; Xiong, C.; Chen, W.; Corso, J.J. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
13. Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; Darrell, T. Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2625–2634.
14. Venugopalan, S.; Xu, H.; Donahue, J.; Rohrbach, M.; Mooney, R.; Saenko, K. Translating videos to natural language using deep recurrent neural networks. *arXiv* **2014**, arXiv:1412.4729.
15. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
16. Yao, L.; Torabi, A.; Cho, K.; Ballas, N.; Pal, C.; Larochelle, H.; Courville, A. Describing videos by exploiting temporal structure. In Proceedings of the IEEE International Conference on Computer Vision, Boston, MA, USA, 7–12 June 2015; pp. 4507–4515.
17. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 2048–2057.
18. Pan, P.; Xu, Z.; Yang, Y.; Wu, F.; Zhuang, Y. Hierarchical recurrent neural encoder for video representation with application to captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1029–1038.

19. Krishna, R.; Hata, K.; Ren, F.; Fei-Fei, L.; Carlos Niebles, J. Dense-captioning events in videos. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italia, 22–29 October; pp. 706–715.
20. Escorcia, V.; Heilbron, F.C.; Niebles, J.C.; Ghanem, B. Daps: Deep action proposals for action understanding. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 768–784.
21. Li, Y.; Yao, T.; Pan, Y.; Chao, H.; Mei, T. Jointly localizing and describing events for dense video captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7492–7500.
22. Venugopalan, S.; Rohrbach, M.; Donahue, J.; Mooney, R.; Darrell, T.; Saenko, K. Sequence to sequence-video to text. In Proceedings of the IEEE international Conference on Computer Vision, Boston, MA, USA, 7–12 June 2015; pp. 4534–4542.
23. Yu, H.; Wang, J.; Huang, Z.; Yang, Y.; Xu, W. Video paragraph captioning using hierarchical recurrent neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 4584–4593.
24. Caba Heilbron, F.; Carlos Niebles, J.; Ghanem, B. Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1914–1923.
25. Manning, C.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S.; McClosky, D. The Stanford CoreNLP natural language processing toolkit. In Proceedings of the 52nd annual meeting of the association for computational linguistics: System demonstrations, Baltimore, MD, USA, 23–24 June 2014; pp. 55–60.
26. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
27. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
28. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–12 December 2017; pp. 5998–6008.
29. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.