Journal of Big Data

RESEARCH Open Access

A novel Multi-Layer Attention Framework for visual description prediction using bidirectional LSTM

Dinesh Naik* and C. D. Jaidhar

*Correspondence:
din_nk@nitk.edu.in

Department of Information Technology, National Institute of Technology Karnataka, Surathkal, Mangalore 575025, India

Abstract

The massive influx of text, images, and videos to the internet has recently increased the challenge of computer vision-based tasks in big data. Integrating visual data with natural language to generate video explanations has been a challenge for decades. However, recent experiments on image/video captioning that employ Long-Short-Term-Memory (LSTM) have piqued the interest of researchers studying its possible application in video captioning. The proposed video captioning architecture combines the bidirectional multilayer LSTM (BiLSTM) encoder and unidirectional decoder. The innovative architecture also considers temporal relations when creating superior global video representations. In contrast to the majority of prior work, the most relevant features of a video are selected and utilized specifically for captioning purposes. Existing methods utilize a single-layer attention mechanism for linking visual input with phrase meaning. This approach employs LSTMs and a multilayer attention mechanism to extract characteristics from movies, construct links between multi-modal (words and visual material) representations, and generate sentences with rich semantic coherence. In addition, we evaluated the performance of the suggested system using a benchmark dataset for video captioning. The obtained results reveal superior performance relative to state-of-the-art works in METEOR and promising performance relative to the BLEU score. In terms of quantitative performance, the proposed approach outperforms most existing methodologies.

Keywords: Attention, Computer vision, Convolutional Neural Network, LSTM, Video captioning

Introduction

Visual captioning, alternatively referred to as video/image captioning, is the process of creating a description/caption for a video/image. The caption/sentence defines the items and actions in the image or video succinctly and precisely. Combining computer vision and Natural Language Processing (NLP) to generate video descriptions was previously considered a difficult task from a vision standpoint. The goal of establishing a correlation between video content comprehension and textual prediction has been the subject of extensive research in recent years [1–5].

The overall video captioning framework is explained in Fig. 1. Establishing a connection between visual stuff and text prediction is a relatively simple task for humans. However, it has been viewed as a particularly difficult problem for machines and a vital component of machine intelligence. The proposed Multi-Layer Attention Framework generates the video descriptions by combining encoder and decoder architectures. Since LSTM encoders are bidirectional, their output is twice as large as the hidden layer. The encoder, which is composed of a BiLSTM, contains 1024 hidden units. In a single operation, the encoder concatenates the BiLSTM's forward and reverse LSTM outputs, resulting in a 2048-byte output. The decoder is developed using a stacked LSTM unit with a single direction. This unit is merged with 2048 hidden units, a 1024-node attention layer, a 256-node embedding, and a dense layer with nodes matching the corpus' vocabulary to achieve better performance. In addition, the decoder consists of fully convolution layers including nodes reflecting the corpus's vocabulary. A notable advantage is that the decoder LSTM has a concealed size that is twice as large as the encoder LSTMs. It has numerous applications, such as video comprehension, video retrieval, and video subtitling.

In recent years, Deep Neural Networks (DNN) [6] have made significant progress in image/video captioning. However, it is not as basic as image captioning, and it cannot be accomplished just through the use of Convolutional Neural Networks (CNN). Ideally, in layered LSTM networks, temporal attention helps more to bridge the gap between video visuals and words to be predicted as output. LSTM networks [7] and attention mechanisms have been used to describe a video in order to improve semantic consistency by capturing the most striking aspects of the visual representation. The work was motivated by a desire to make a significant contribution to the field of LSTM application. Using Bidirectional LSTM [BiLSTM] [8–10] in the encoding step of the framework has resulted in improved overall performance of the framework.

Attention is a cutting-edge breakthrough in the deep learning realm. It has resulted in significant advancements in a variety of domains such as machine translation, visual captioning, question answering, and so on. To perform a soft selection over source elements based on their categorical distribution, we create multilayer attention in a hidden layer and apply it to the source elements. The attention on the framework has made a substantial contribution to its overall performance.

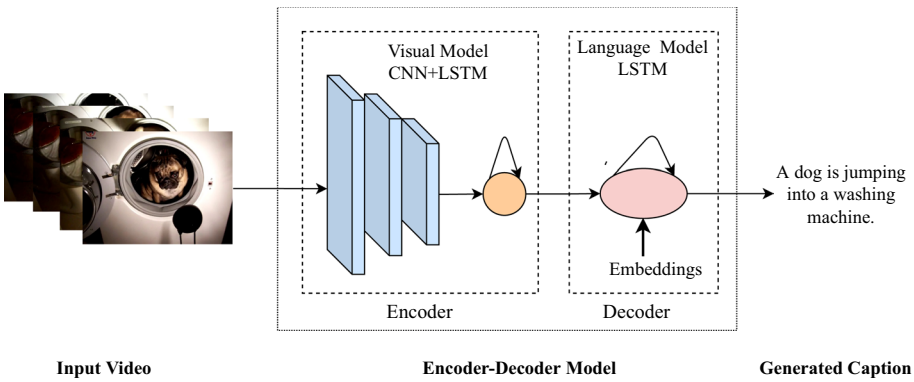

Fig. 1 The overview video captioning framework

At the start of the procedure, we utilised the VGG-16 [11] model to extract contextual details from the frames of a clip. The work in this proposed framework was inspired by the work reported in [8] and led to the usage of VGG-16 in the proposed framework. In this case, the spatial 2D CNN highlight vectors are encoded using an LSTM-based visual encoder. The contextual attention mechanism employs the dynamic weighted sum of neighbouring spatial 2D CNNs that contain vectors as the input to the LSTM decoder's multiple layers.

The proposed framework results are obtained by coupling a state-of-the-art notion called attention with a variation of the Recurrent Neural Network (RNN), particularly the LSTM network, which is effective in learning long-term dependencies. In order to provide accurate descriptions for video clips, video captioning uses a combination of visual content interpretation and NLP. The multimedia community has been studying the topic of video captioning extensively. However, most present techniques significantly rely on static visual information or capture just a portion of local temporal knowledge, making it difficult to adequately describe motions from a global perspective. A multi-layer bidirectional long short-term memory (BiLSTM) structure is proposed to describe video's temporal dependencies, which investigates both forward and backward temporal information in the entire sequence of video frames. To fully leverage the video clip's bidirectional global temporal structure, we create a joint model that combines a forward pass LSTM, a backward pass LSTM, and CNN features. Furthermore, to enhance our model's focus on semantic and relevant information, we add an attention mechanism to the decoding stage of the multi-layer unidirectional LSTM. As a result, the video's one-way global representation and one-way local concentration could be useful to our model. Captions for given videos can be accurately described or predicted thanks to the framework's extensive use of hidden semantic information in videos and texts.

The motivations behind this result-oriented approach are manifold. Modeling the temporal dependencies of video content, the majority of publications employed RNN for video content description. Nonetheless, the LSTM and GRU have proven useful breakthroughs that provide excellent performance outcomes. Combinations of bidirectional and unidirectional LSTM have been applied to both the encoding and decoding portions to maximise the potential of forward and reverse networks. In addition, the BiLSTM structure fully examines both forward and backward temporal information over the whole video frame sequence. In addition, researchers discovered that the visual attention mechanism facilitates the efficient comprehension of visual content by machines. The suggested research makes use of temporal soft attention in a hierarchical manner to focus on certain phrases and contexts in videos.

The following are the most significant contributions made by our present research.

- The proposed framework makes use of a novel multi-layer BiLSTM encoder and a multi-layer unidirectional decoder.
- Both the encoder and decoder units employ two layers of temporal soft attention. This emphasis on the complete global view of video segments adds additional representational features.
- Additionally, to ascertain the superiority of the proposed framework, three variants of the models are examined.

- Additional trials on two benchmark video captioning datasets demonstrate our proposal's superiority over existing standard methodologies.
- The proposed framework is evaluated with performance metrics like BLUE and METEOR.

The remainder of the paper is divided into the following sections. The section on related works explores the study of existing works. The proposed methodology section explains the proposed framework's multiple components. The experiments, results and discussion section provides in-depth explanations of the conducted experiments and their outcomes for video captioning. Finally, the Conclusion section summarises the proposed work and makes future research directions.

Related works

Determining how to describe an image or video has been the subject of extensive research over many years. The efficiency of the model is achieved by using deep learning methods and other modules. Based on the research done in this field, the approaches used can be put into two main groups: bottom-up approaches and top-down approaches. The articles in the bottom-up approach [12–16], the various aspects of the video are taken and analysed and compared against, and once a pool of terms has been generated, these terms are merged to obtain words/sentences that are used to generate relevant descriptions and captions for the video. The top-down approaches [17–20], use a differentiated approach, attempting to generate sentences from a expression that encompasses all of the video's spatial and temporal aspects.

The article [12] introduces a comprehensive, data-driven method for creating natural-language definitions of short videos by selecting the most effective subject-verb-object triplet for describing realistic MSVD videos. By utilising knowledge extracted from huge corpora to assess the likelihood of alternative SVO combinations, they improve the capacity to identify the optimum triplet for describing a video and develop sentences that are chosen by both machine and human evaluation. In their strategy, linguistic expertise dramatically enhances activity detection, particularly when the distributions of the training data and the test data differ greatly.

The authors of the paper Ref. [13] developed a new method, Hierarchical Recurrent Neural Encoder (HRNE), for constructing video representation with a focus on temporal modelling. The proposed HRNE is more capable of video modelling than existing approaches because (1) HRNE reduces the size of input data flow and leverages temporal structure over a wider range and at a higher level; (2) HRNE adds more non-linearity and flexibility; and (3) HRNE limits temporal transitions at multiple levels of granularity.

In the study Ref. [14], researchers describe a new strategy, a Factor Graph Model (FGM), to determine the optimum subject-verb-object-place description of a clip by merging visual and linguistic information. Additionally, the model incorporates scene (location) information.

The article [17] offered an innovative strategy for video description. The authors develop descriptions using a sequence-to-sequence model, in which sequential reading of frames is followed by consecutive generations of words. This enables for variable-length input and output while simulating temporal structure simultaneously.

The authors of Ref. [18], introduced a unique three-dimensional convolutional neural network capable of capturing local fine-grained motion data from consecutive frames. They propose employing a temporal visual attention mechanism that acquires the ability to pay attention to subsets of frames in order to acquire global temporal structure. Lastly, the two proposed methods naturally integrate into an encoder-decoder neural video caption generator. In Ref. [19], the construction of a novel bidirectional LSTM (BiLSTM) network for video captioning is proposed. Specifically, they construct joint visual modelling to investigate bidirectional global temporal information in video data fully by merging a forward LSTM pass, a backward LSTM pass, and CNN's characteristics. To improve subsequent sentence synthesis, the acquired visual representations are fed as initialization to an LSTM-based language model.

The article [20] presents a unified framework called aLSTMs, a semantically consistent attention-based LSTM model. First, they employ the Inception-v3 neural network, an expanded version of GoogleNet, to extract more significant spatial characteristics. In order to take advantage of temporal information, they developed a one-layer LSTM visual encoder to capture the spatial 2D CNN feature vectors. The model also incorporated an attention mechanism that uses the dynamic weighted sum of local 2D CNN feature vectors as input for the LSTM decoder. They integrate multi-word embedding and cross-view approach to project the generated words and visual elements into a shared space in order to bridge the semantic gap between videos and their related texts.

In Ref. [14, 17] used a LSTM network, to simulate the overall temporal structure of the video sample. However, these approaches did not take advantage of bidirectional global temporal structure, which may benefit from both previous and future video frames.

The article [20] focused on a LSTM network and how to model the global temporal structure of the whole video snippet. But these methods didn't take advantage of global temporal structure that goes both ways. But the article [19] used BiLSTM to take advantage of the benefits of forwarding and backward pass.

Stacked Multimodal Attention Network (SMAN) is a revolutionary video captioning paradigm proposed in Ref. [21]. It adopts extra visual and textual historical information as context features during caption generation, utilized a layered architecture to process distinct data gradually, and employs Reinforcement Learning and a coarse-to-fine training technique to further enhance the obtained results. Captioning can be improved by using a semantic and syntax-guided hierarchical attention network (SHAN) [22] to incorporate visual and sentence-context elements. An object-scene relational graph model [23] is created to convey the association characteristics based on the object detector and scene segmenter in order to address key fine-grained semantic qualities and video variety. An encoded graph neural network enhances visual qualities by encoding the graph.

Recent research has demonstrated that the soft attention mechanism is highly effective and has attracted increased interest from the computer vision field. The authors of Ref. [8] used this attention technique to concentrate on specific information and adjust the sentence's prediction to the associated video. However, there is a lack of diverse levels of attention, which might be addressed.

The other major field of research, particularly at the early stages, was sequential modelling, which understands features and other parameters to correlate visual

contents vector and a textual sentence into a semantic vector space and investigates distribution in the merged space. The article [24] discusses how two stacked RNNs are used to decode video frame information into a mapping space that is then utilised for captioning.

The combination of an image model with CNN and a language model with LSTM architecture has gained popularity in the research community and has affected video captioning with practically excellent results. These combinations virtually always demonstrated superior performance on a variety of standard datasets. Both the feature extraction and word generation methods are critical for the quality of the input video's description.

In Ref. [25], the soft attention module selectively selects the most pertinent frames and the proper location of objects within each frame. Additionally, the authors proposed an attention unit that would prioritise the most comparable phrases in order to exploit more precise language descriptions. For visual captioning, an approach based on hierarchical LSTMs with two-level abstraction was presented in Ref. [26]. The authors of Ref. [27] demonstrated a two-tiered approach to determining the context of words in captions. This method is superior to the other because it detects long-distance text sequence dependencies and is faster to compute. To ensure the semantic compatibility of the sentence description and the visual content of video [28], suggests mapping the visual and textual properties into a joint space using an attention mechanism with a local two-dimensional encoder and LSTM decoder.

The RNN is intended to handle grouping tasks such as machine interpretation, language interpretation, and music composition [29] which persistently saves data about previous activities through the use of critical associations. Two distinct master networks with linear and logarithmic combinations are combined using a bidirectional RNN for language recognition, which extensively utilises the whole dataset of Schuster and Paliwal [30]. In comparison to bidirectional RNN, the BiLSTM is a type of upgraded RNN with a long history of use in the domain of natural language processing.

Karpathy et al. [31] have also employed bidirectional RNNs in video captioning, where they are used to learn the relationship between video frames and English like sentences and to add word embedding. Ullah et al. [32] use a BiLSTM and CNN features to extract activities from videos. The result in Ref. [32] established that bidirectional ones outperform unidirectional ones in sequence processing.

The research [33] analyses the effect of attention on spatial and temporal features and contributes to the effectiveness of their deep network P3D ResNet and 2D-CNN. Additionally, they examine the use of language models such as LSTMs and Sentence Transformers in video captioning. RNNs make another contribution to dynamic nonlinearity by mapping input sequence data to output sequences.

Research methodology

The proposed work's overall objective is to generate English language descriptions of the video content. This section will discuss the proposed approach for video description generation using LSTM, BiLSTM, and Multi-Attention. Following that, the technical concepts underlying the proposed methodology are described.

Preprocessing unit

- *Video preprocessing* In order to lessen the computational cost, we take 30 frames from each video that are evenly spaced. In this step, the VGG-16 model is used to retrieve the features from video frames, which is then fed into the encoding unit to provide a global view of the videos.
- *Text preprocessing* In order to clean up the corpus, deleting any unnecessary spacing and special symbols that were there. We eliminate sentences with fewer than three words and more than 30 words since more than 90 percent of the sentences have lengths larger than three words and fewer than 30 words, respectively. The $< BOSs >$ and $< EOS >$ tokens are added at the beginning and end of each phrase, respectively, to mark the beginning and end of sentences. When a batch of these sentences is formed, a token $< pad >$ is added to ensure that all of the sentences are of the same length, which increases the computational speed of the batch.

Inspired by the baseline [8], this article presents a total of four deep learning models, one of which being the base model. The following paragraph summarises and lists all of the models.

- Base line model: This design is backed up by research outlined in Ref. [8]. The encoder and decoder, respectively, are made up of one BiLSTM and one unidirectional LSTM, as indicated in the Fig. 2.
- Base model with batch-normalization: A batch normalisation layer at the encoder's output and another batch normalisation layer at the decoder's LSTM output were incorporated in this model.
- Stacked LSTM: Two BiLSTMs are layered together for the encoder in this architecture, while two unidirectional LSTMs are stacked together for the decoder as depicted in Fig. 3. The attention layer makes use of the encoder's second BiLSTM output at every time interval, as well as the hidden state of the decoder's second LSTM. However, the hidden states of both language model LSTMs were initialised with the hidden states of the visual model.
- Multi-layer attention model:The encoder and decoder are constructed by stacking two BiLSTMs for the visual model and two unidirectional LSTMs for the language model, respectively. The following step is to use the output of the first BiLSTM of the visual encoder at each time step, as well as the hidden unit of the first LSTM of the language model, for the first attention layer. The second attention layer is formed using the outcome of the second BiLSTM of the visual encoder at every sampling interval, as well as the hidden unit of the second LSTM of the word embedding. To preconfigure the hidden states of either the LSTMs in the language model, they must first be populated using the visual encoder's hidden units. This proposed Multi-Layer Attention Framework for video description generation is depicted in Fig. 4.

The fundamental distinction between the Baseline and Stacked LSTM models is that the Stacked LSTM model's encoder and decoder are built of two layered LSTMs, as

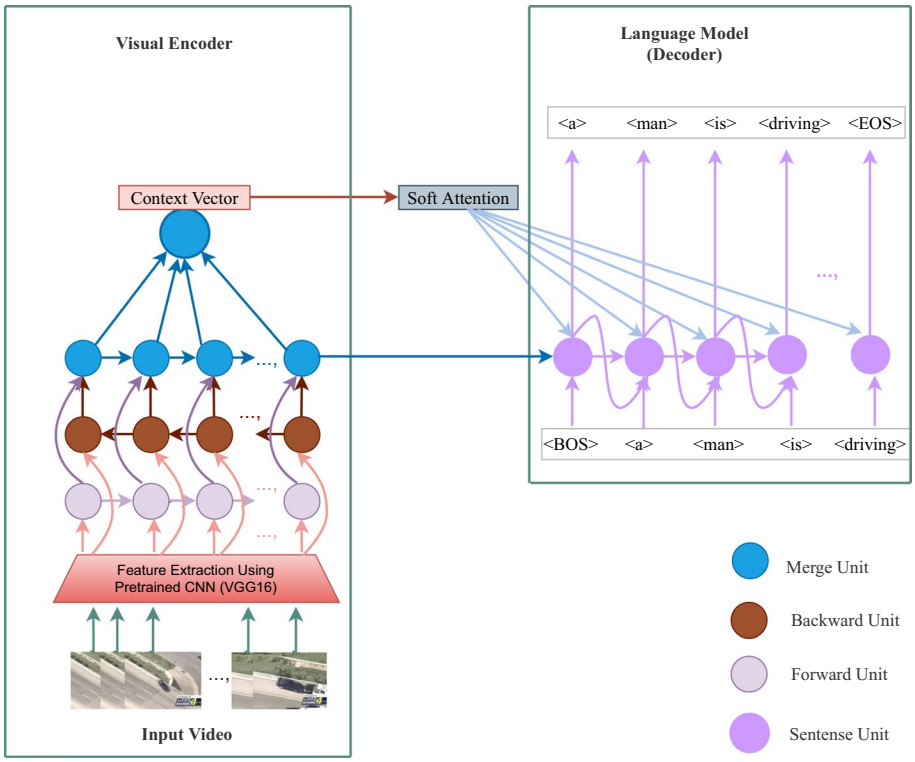

Fig. 2 Base Model Architecture [8]

previously described and illustrated in Fig. 3. Similarly, the second LSTM output is considered while building the soft attention mechanism in the encoder, and it is employed to generate one word at a time when developing the word output mechanism in the decoder. The Stacked LSTM Model is twice as large as the Baseline due to the presence of two LSTM in both the visual encoding and language model stages. It is imperative to highlight between the Stacked and Multi-Layer Attention Models because the Multi-Layer Attention Model incorporates an additional attention layer interconnecting the encoder's first BiLSTM and the decoder's first LSTM, as depicted in Fig. 4.

Architectures for encoder and decoder are used in conjunction to create the framework under consideration. There are 1024 hidden units in the encoder, which is made up of a BiLSTM. The encoder generates a 2048-byte output since this concatenates the BiLSTM's forward as well as backward LSTM outputs in a single operation. Since the encoder and decoder work together to initialize the decoder with video representation. Additionally, we use an original CNNs feature to continuously merge forward and backward passes with a BiLSTM as Merge Unit (MU) to inject the final hidden state and memory cell into the language decoder as a global video representation. In light of the attention method, we assume that the context set to which the sentence generator pays attention, contains all of the output states of the merging layer.

The decoder is constructed using a single direction layered LSTM unit. To achieve higher performance, this unit is merged with 2048 hidden units, a 1024-node attention layer, a 256-node embedding layer, and a fully connected layers with nodes

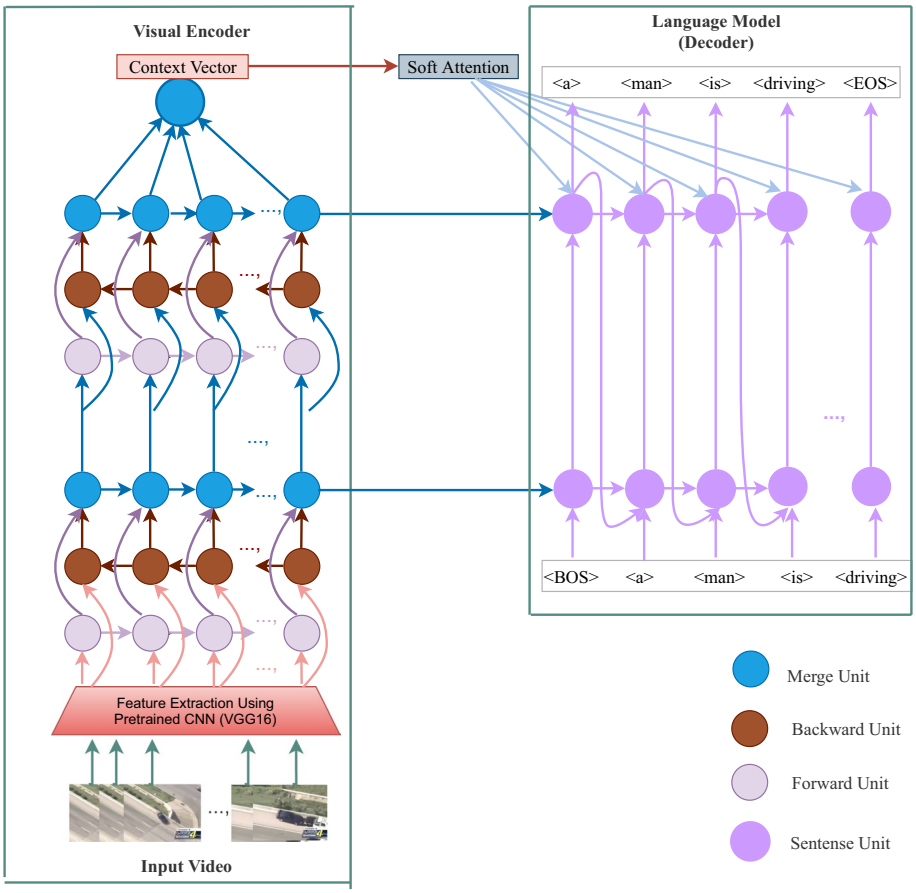

Fig. 3 Stacked LSTM Model Architecture

matching to the vocabulary of the corpus. The decoder LSTM has a concealed size that is twice as large as the encoder LSTMs, which is a significant advantage.

LSTM encoders are bidirectional, which means that their output is double the size of their hidden layer, as explained above. Utilizing a decoder of above mentioned size enables us to take advantage of the encoder's hidden states in the LSTM decoder, which is a considerable advantage. This permits the encoder's overall video content interpretation to be propagated to the decoder's global video representation.

The encoder BiLSTM is used to generate a global representation of the input video from each video that has been processed once it has been obtained after preprocessing. This data is kept in order to make it available to the decoder's attention layer at each and every time step of the usual decoding stage. When a end of the time step is reached, the attention unit delivers a context vector that contains the encoder output and the decoder's hidden state. When a sentence is input, the decoder goes over each word and generates the following word. Each word is fed to the decoding embedding unit at the input, and its output is combined with the context vector received from the attention unit. This output is fed to the LSTM decoder, which decodes it. The outcome of the LSTM is sent to a fully connected layers, which generates a vector with a length equal to the vocabulary size of the corpus and including information about the next term.

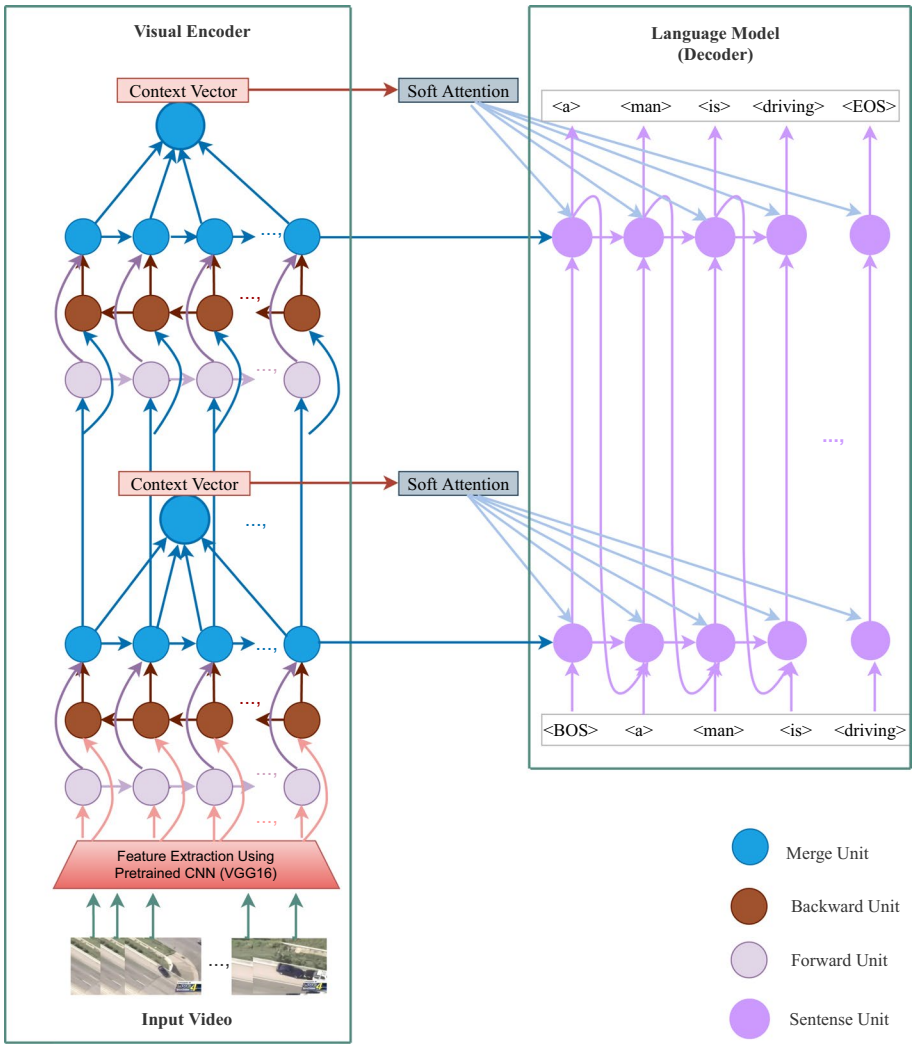

Fig. 4 Proposed Multi-Layer Attention Framework

Sequential model using LSTM-based Neural Network

To use a single input sequence, conventional RNNs can theoretically take account of arbitrary long-term relationships in word sequence. When utilising LSTM units as RNNs, the vanishing gradient problem is partially solved due to fact that LSTM units ensure gradients to continue to flow unchanged or unmodified. Initially, a video clip $V = (F_1, ..., F_N)$, is used, with F_t denoting the video's t^{th} frame as the initial starting point for the captioning task. In this case, the primary goal is to encode video with words and express the result as a feature vector $V_{feature}$. The repeating nature of each frame must be considered in order to emphasise the time dependence of the frames' content. The RNN variation maps the input word sequence $X = (x_1, ..., x_t)$ to an output word sequence $Z = (z_1, ..., z_t)$, which can be expressed as

$$h_t = \phi(W_{hx}x_t + W_{hh}h_{t-1}), \tag{1}$$

$$z_t = \psi(W_o h_t), \tag{2}$$

Where, F_t: which symbolises the t^{th} frame. $V = (F_1, ..., F_N)$:denotes the contextual feature vector. W_{h*}: Weights that are related to previously hidden states / the present input. W_o translates the hidden states between the hidden and output spaces. h_{t-1} and h_t: represent the RNN's hidden states at $t-1$ and t, respectively. ϕ and ψ represent nonlinear, two functions, respectively.

While traditional RNNs suffer from the gradient vanishing or explosion problem, an upgraded RNN stores information in a memory cell and uses numerous control gates to have read-write operation from and to the memory unit or cell respectively, resulting in improved performance when leveraging extremely long temporal dependency relationships. Refer to Fig. 5 for an illustration of the core LSTM architecture, and all of the gate information can be logically stated as follows:

$$i_t = \sigma(W_{ix} x_t + W_{ih} h_{t-1}), \tag{3}$$

$$f_t = \sigma(W_{fx} x_t + W_{fh} h_{t-1}), \tag{4}$$

$$o_t = \sigma(W_{ox} x_t + W_{oh} h_{t-1}), \tag{5}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \phi(W_{cx} x_t + W_{ch} h_{t-1}), \tag{6}$$

$$h_t = o_t \odot \phi(c_t), \tag{7}$$

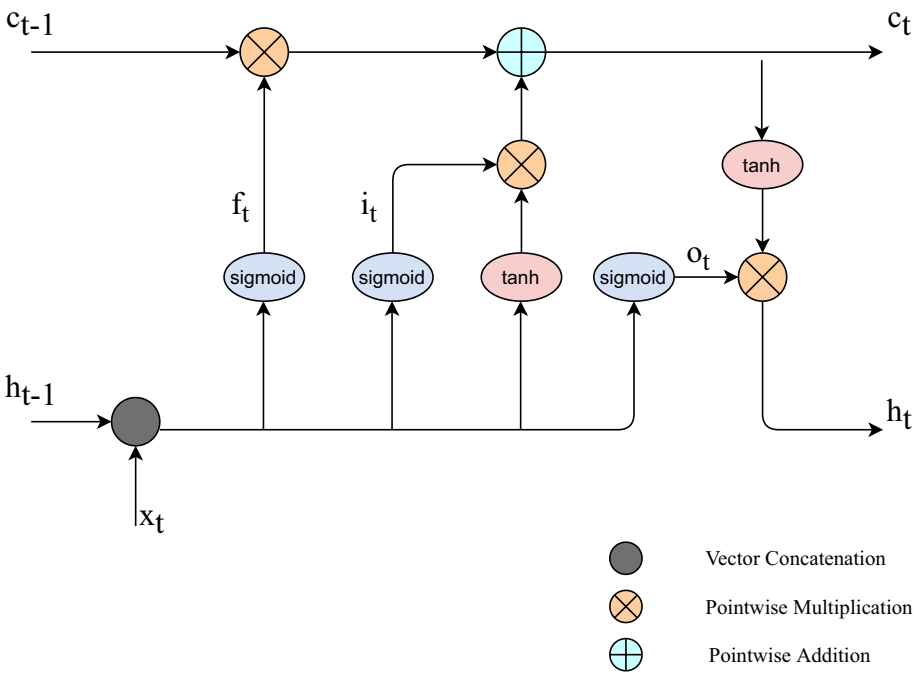

Fig. 5 LSTM architectural preview

Where, W_{*h}: Weights that relate each gate in the LSTM to previous hidden states. W_{*x} :Weighing units that connect the current input to each gate. σ: depicts the sigmoid nonlinear activation functions. ϕ: depicts the hyperbolic tangent nonlinear activation functions. $\odot$: represents the operation of element-by-element multiplication.

Bidirectional LSTM

CNNs have entirely independent inputs and outputs, but in some cases, the model may need to recollect prior meaningful words to choose the subsequent relevent word. For example, you might be watching a video clip and pausing to estimate the finish; your guess will be based on already watched portion of clips and what interpretation has come to mind so far. RNN recalls the previous event and tries to predict the next word in this way. This way it tunes to solve the CNN problem by introducing a hidden units as a layer into the network.

An LSTM recalls every piece of information over the course of time, just as it remembers prior inputs. It is advantageous in the prediction of time series. Bidirectional RNNs connect two RNNs together, allowing them to provide information on both the forward and backward sequences. LSTM provides stronger sequence processing capabilities and has the capacity to detect lengthy dependencies in sequences. LSTM-based networks are used to analyse the temporal feature for video and, the framework uses them to do so. It does this by mapping the video-level activity into the language model, which results in word-for-word video descriptions.

Whilst employing unidirectional LSTM, only past data may be used as inputs; thus, only previous data can be preserved. Alternatively, when employing BiLSTM, inputs can be processed in either direction: forward or backward. This strategy is far more effective at any moment in time, because it allows you to obtain knowledge from both the future and the past by combining two latent states.

In BiLSTM, contextual information is processed in both the forward and backward directions, allowing for the retention of information from both the past and the future. According to recent study, bidirectional RNNs produce better results when relying on sequential voice recognition processing and image captioning for a lengthy period of time. The Bidirectional LSTM architecture used in our proposed framework depicted in Fig. 6.

Temporal attention

Soft attention is a global attention in which weights are placed lightly across all visual patches. Therefore, soft attention assesses the complete source image. Suppose we are observing the face of a woman. We may concentrate on her face and render its features with precision, but we may also observe her clothing and hair. In other words, humans are able to "focus" our visual attention on a tiny region without losing awareness of other visual regions. Specifically, we simply modify the "weights" of our visual attention for each visual field, assigning high weights to the areas of focus. Xu et al. [34] segmented an image into multiple patches and employed a spatial attentional mechanism to determine which patches should be "focused" as each word is formed for image description. The soft attention can also be viewed as temporal attention.

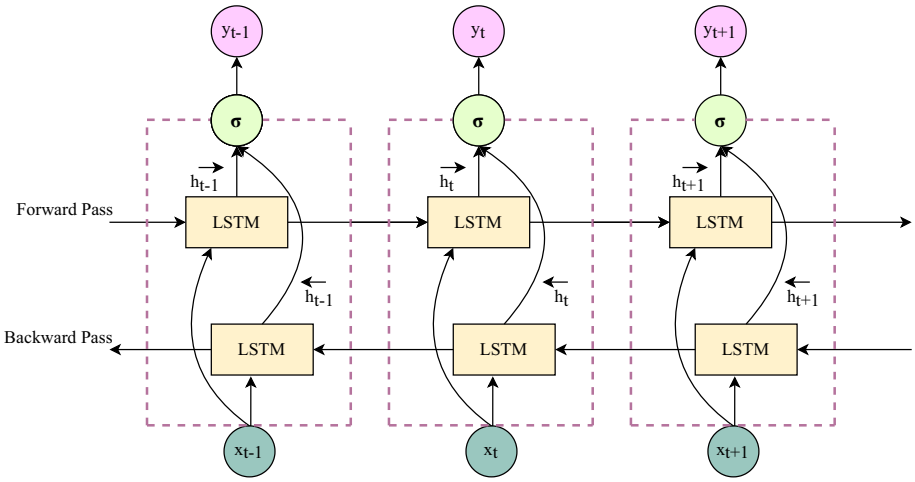

Fig. 6 Bidirectional LSTM architecture

The study in recent past on neural deep learning network has made extensive use of the attention mechanism, particularly in domains requiring vision, such as video/image captioning and machine translation. The basic notion is to increase the emphasis on a specific section of an image or video frame. A mechanism of attention that placing a greater emphasis on essential or crucial video frames with objectivity and their associations such as human actions, as opposed to its spatial mechanism, which promotes on the image's more semantically significant components.

Temporal attention can be viewed as a context set of visual elements with a window of visuals. Depending on the context, these visual features are referred to as regions or frames. At each discrete time step, the attention vector in Eq. (8) can be created in conjunction with the dynamic weights for each visual element. The value in Eq. (9) is appropriately accommodated by the dyanamic weights. For each visual element along the last concealed state shown in Eq. (10), a relevancy score is produced.

$$sa_t = \sum_{i=1}^{m} \alpha_i^t vc_i, \tag{8}$$

$$\sum_{i=1}^{m} \alpha_i^t = 1, \tag{9}$$

$$\gamma_i^t = W_{rel} tanh(W_a vc_i + U_a h_{t-1} + b_a), \tag{10}$$

Eq. (11) used to standardise the acquired relevance ratings.

$$\alpha_i^t = exp(\gamma_i^t) / \sum_{j=1}^{m} exp(\gamma_j^t), \tag{11}$$

Where,

FOV: Field Of View. vc_i: Represents i^{th} element of context set. $VC = (vc_1, ..., vc_m)$: Denotes visual context set. α_i^t : Signifies dynamic activation weights for each element in context set. sa_t : Generated attention vector. γ_i^t : Context set relevance score. W_{rel} : relevance parameter for Context set. W_a : Context element parameter. U_a : Hidden state learning parameter.

The approach, soft attention attempts to replicate the attention allocation cycle for a given field of vision. By integrating forward and backward passes with temporal attention and applying temporal attention in the process, the current framework constructs sentences word by word in two phases. The approach produces a context set for each specific situation by utilising CNN highlights of edges and other latent information states in the merging layers. Moments after the production of a word, temporal attention directs the language model's focus to explicitly wordly locations that are more semantically significant. When the input word sequence is combined with the output, it is advantageous to consider the attention vector with the input, as illustrated in the base model, Fig. 2.

Batch normalization's mathematical model

When training a DNN with multiple layers, the outcomes may differ due to factors such as the learning algorithm's design and the initial random weights. Due to the fact that the weights are updated after each mini-batch, the distribution of inputs to the network's deep layers differs with each mini-batch. They can make it more challenging for the model to acquire new skills by following a moving object. A deep neural network's "internal covariate shift" refers to a change in the proportion of inputs to layers due to the shift in the internal covariate dispersion of a network. Large neural network models that have a large number of inputs are standardised batch-by-batch using a technique known as Batch Normalization. This strategy significantly decreases the number of epochs necessary for training while simultaneously stabilising the model's learning process. During training, batch normalisation can be accomplished by computing the mean and standard deviation of each input parameter for each mini-batch. Finally, these results are employed to restore the network's representational capacity; a transformation logic is established. The mini batch mean is calculated using Eq. (12),

$$\mathrm{E[x]}_B = \frac{1}{m}\sum_{i=1}^{m} x_i, \tag{12}$$

where x_i is values of x over a minibatch, $B = x_{1....m}$.

The mini bacth variance can be defined with help of Eq. (13),

$$\mathrm{Var[x]}_B^2 = \frac{1}{m}\sum_{i=1}^{m} (x_i - \mathrm{E[x]}_B)^2, \tag{13}$$

Now, for a layer with d-dimensions, $x = (x_1 ... x_d)$, each dimensions of its input can be normalized using, Eq. (14),

$$\widehat{\mathrm{x}}_i^k = \frac{\mathrm{x}_i^k - \mathrm{E}[\mathrm{x}]_B^k}{\sqrt{\mathrm{Var}[\mathrm{x}]_B^{k^2} + \epsilon}}, \tag{14}$$

where $k \in [1, d]$ and $i \in [1, m]$. The ϵ is an arbitrary small constant for numerical staility. The final transformation is logically defined in Eq. (15),

$$\mathrm{y}_i^k = \gamma^k . \widehat{\mathrm{x}}_i^k + \beta^k, \tag{15}$$

The γ and β are learnable parameter during the optimization process. By default, γ elements are set to 1 and β elements to 0.

Multi-layer attention

Following preprocessing in this proposed framework, each video is transmitted onto the encoder BiLSTM, which produces a global representation of the original video. During each time step, the encoder's output is kept in order to be fed into the decoder's attention layer [35]. During each time step, the attention unit receives the encoder output from the decoder as well as the hidden state and returns a context vector. When a sentence is input, the decoder takes each word in turn and produces the following word in the sentence.

The framework proposes adding two layers of LSTM and two attention layers shown in Fig. 4. The outcome of the framework is discussed in the result section. The intent behind the framework as based on earlier research by Schuster et al. [30], Karpathy et al. [31], Ullah et al. [32]. The performance of bidirectional LSTMs is always significantly superior than the performance of unidirectional models, which has been found in a variety of fields such as image captioning, speech recognition, and action recognition. Adding another layer of attention allows for a more in-depth examination of the results of both layers of attention. The addition of two layers of attention to the framework can help it to optimise the results even more effectively.

Experiments, results and discussion

Dataset

- *MSVD* This standard dataset called Microsoft Video Description (MSVD) [36] corpus, is comprised of 1970 videos and additional 80,000 captioning sentences. Each video clip is between eight and twenty-five seconds long. Each sentence has roughly seven words, and each video contains approximately 43 sentences total for the duration of the video. The dataset contained 1970 videos, which we utilised in our research. We divided the entire dataset into two groups: 80% for training and 20% for testing.
- *MSR-VTT 10K* The MSR-VTT [37] corpus has ten thousand video clips with twenty descriptive text for each clip. The dataset categorises videos broadly, including "music," "TV shows," and "tourism." Each clip lasts between 10 and 30 s. The dataset contains about 20,000 unique words, with an average of around ten words per description. The framework was built using a dataset split into 6513, 497, and 3000 rows for train, validation, and test, respectively.

Experiments

Implementation details

PyTorch was used to create the framework. The work took into account both the dataset video-sentence as well as a sample. When an input sentence sequence is fed further into the decoder, a new word is produced alongside the original word. The proposed models were optimised using an Adadelta [38] and initial learning parameters. The additional hyperparameters are defined as follows: ρ with a value of 0.9 and ϵ with a value of 10^{-6}, respectively. In this instance, a mini-batch of size 64 was employed during training. The framework trained iteratively until the best results.

- **MSVD** To train these models using the dataset, the framework was created using open source Google Colab. Each epoch took 570 s, 800 s, and 1040 s, respectively, for the Base Model, the Stacked LSTM model, and the Multi-Layer Attention model.
- **MSR-VTT** To train these models using the dataset, the framework was created using open source Google Colab. Each epoch lasted 1156, 1520, and 1940 s, respectively, for the Base Model, Stacked Model, and Multi-Attention Model.

A word was constructed by initialising the global video interpretation with the $< BOS >$ token and then outputting the following word using the decoder's output.

The framework includes the previous output word as an input word for that video's caption until the decoder generates a $< EOS >$ token or the maximum count of words possible for that video's caption. Regardless of whether the model accurately captions a video, if the amount of words in the reference sentences exceeds the number of words in the caption, the model's BLUE [39] and METEOR [40] scores are dropped, respectively.

To obtain the required results, the framework determines the maximum count of words to output as the average count of words in the reference collection for that individual video clip.

Results and analysis

The metrics BLEU and METEOR are two of the most commonly used in video/image captioning. METEOR performance metric pulls from the source documents and evalautes the precise steming, paraphrasing, and synonym matching. It makes use of the WordNet database [41] and determines similarity scores at sentence level, allowing it to capture all semantic components of a sentence.

When comparing a candidate sentence to a large number of reference sentences, BLEU-n employs modified precision, which is the ratio of candidate n-grams in the corpus to the total number of candidate n-grams. The BLEU score evaluates textual and lexical coherence but does not evaluate semantic coherence.

Because METEOR is more robust than BLEU, we used METEOR as our key metric and BLEU as a secondary metric to evaluate model variations in our study.

Table 1 contains information about the performance of all the models on the MSVD and MSR-VTT datasets. The experimental observations demonstrated that by adding

Table 1 Performance of proposed models on MSVD and MSRVTT Dataset

Model	MSVD					MSR VTT				
	B@1	B@2	B@3	B@4	METEOR	B@1	B@2	B@3	B@4	METEOR
Base model	66.01	49.42	38.69	27.19	48.14	57.75	37.49	29.50	16.05	36.25
Base model with BN	62.07	40.28	27.07	16.61	39.30	63.09	38.84	26.99	14.02	35.82
Stacked LSTM	67.49	51.98	41.90	31.23	49.19	58.18	41.41	32.02	17.61	37.88
Multi-layer attention(Proposed)	70.50	56.62	49.60	33.07	51.77	60.33	43.72	34.12	19.61	39.47

Batch Normalization to the Baseline Model, there is a slight decrease in the performance on the test set on all evaluation metrics except B@1 and B@2 scores on MSR-VTT datasets. Stacked LSTM Model performance exceeds both the Base Model and the Base Model with (Batch Normalization). It regularly generates high-quality results. When both models are trained to the exact count of epochs, it outperforms the Base Model considerably. In comparison to the Base Model, the Stacked/layered LSTM Model is twice as large and hence capable of incorporating more detailed semantics from the video clips than another two models.

The Fig. 7 depicts the training loss curve for our developed framework using the MSVD dataset. This demonstrates that the model loss dropped steadily and stabilised around the 50–60 range epochs. The best findings have been selected and are given in Table 1. The proposed Multi-Layer Attention framework outperforms the other three models in terms of overall performance. However, even though the proposed framework size is the same as that of the Stacked LSTM model and the training loss of both models is virtually the same, the proposed framework outperforms both models on the test datasets, demonstrating that the first attention layer has an impact. When comparing the Stacked LSTM Model to the Multiple Layers of Attention Model, the model benefits from the additional layers of attention.

The Fig. 8 displays the MSR-VTT dataset's training loss curve. This illustrates that the model loss decreased gradually and stabilised around the 40–50 epoch range. The performance results of all the model measured on the MSR-VTT dataset is also summarised in Table 1. The proposed Multi-Layer Attention framework outperforms all three of the other variations in terms of performance by a wide margin. For this reason, relevant words have been focused by soft attention at each layer, allowing them to be predicted.

All of the variants in our proposed framework were evaluated using the performance metric METEOR, which was then compared to some of the existing video captioning state-of-the-art works on both datasets, as shown in Table 2. As a result of semantic attention being paid at both the encoding and decoding stages in our implementations,

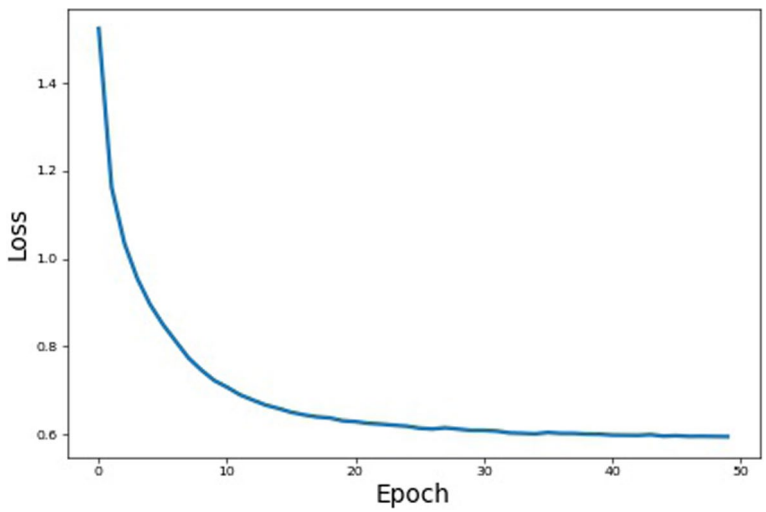

Fig. 7 MSVD Training Loss at each Epoch

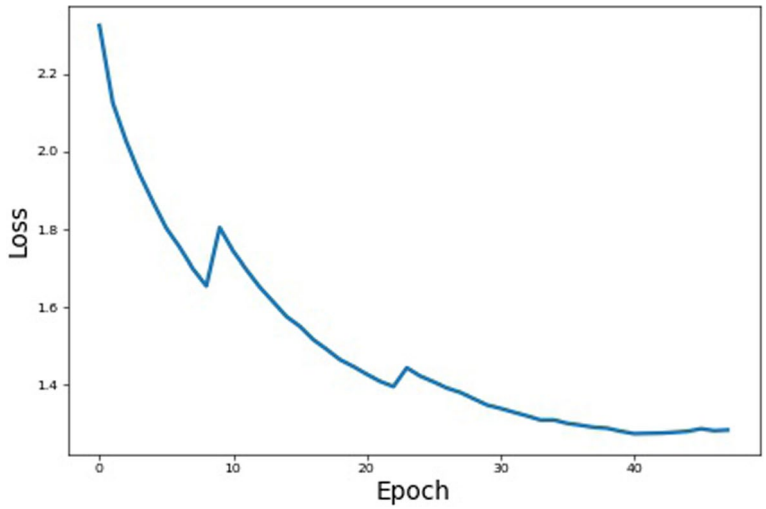

Fig. 8 MSRVTT Training Loss at each Epoch

Table 2 Performance comparison of METEOR score with state-of-art-methods

MODEL	METEOR	
	MSVD	MSRVTT
LSTM [42]	26.9	23.4
LSTM-E[VGG] [42]	29.5	–
LSTM-E[C3D] [42]	29.9	–
MM-VDN [43]	29.0	–
LK [44]	30.3	–
S2VT-unidirectional [17]	29.6	25.2
S2VT-bidirectional [17]	29.7	25.6
S2VT-reinforced [17]	29.9	25.9
S2VT-VGG [17]	29.2	–
S2VT-VGG+Flow (Alexnet) [17]	29.8	–
DVWA-uni [8]	29.6	25.7
DVWA-BiLSTM [8]	29.8	26.1
DVWA-ReBiLSTM [8]	30.3	26.2
DVWA-uni SA [8]	30.2	25.9
DVWA-BiLSTM SA [8]	30.5	26.2
DVWA-ReBiLSTM SA (shortcut) [8]	30.7	26.4
DVWA-ReBiLSTM SA (attention) [8]	30.9	26.6
Base Model	48.14	36.25
Base model with BN	39.30	35.82
Stacked LSTM	49.19	37.88
Multi-layer attention (Proposed)	51.57	39.47

all of the variations on both standard datasets that were used exceeded all of the other findings. The adoption of BiLSTM has also been shown to have a considerable impact on the performance of the models in question.

Table 3 shows the performance metric B@4, of all the variations in our proposed framework. Also, it is compared with some of the existing state-of-the-art video

Table 3 Performance comparison of BLUE-4 Score[B@4] with state-of-the-art-methods

MODEL	B@4	
	MSVD	MSRVTT
STAT [45]	52.0	39.3
SpatioTempo [46]	47.9	38.3
LSTM [42]	31.2	–
LSTM-E [ALEX] [42]	38.9	–
LSTM-E [C3D] [42]	41.7	–
FGM [15]	13.68	–
LSTM-YT [24]	31.19	–
MP-LSTM [24]	33.3	–
Base model	27.19	16.05
Base model with BN	16.61	14.02
Stacked LSTM	31.23	17.61
Multi-layer attention (Proposed)	33.07	19.61

Table 4 Performance score with tuned parameters

MODEL	MSVD					MSR VTT				
	B@1	B@2	B@3	B@4	METEOR	B@1	B@2	B@3	B@4	METEOR
Multi-layer attention (Proposed) without dropout	70.50	56.62	49.60	33.07	51.77	60.33	43.72	34.12	19.61	39.47
Multi-layer attention (Proposed) with dropout	67.79	52.29	45.36	30.36	50.59	58.02	41.30	31.82	16.99	38.35

captioning works on both datasets using B@4. To the best of our understanding, the proposed framework surpassed practically three of the existing video captioning works on MSVD that have been listed. In contrast to this, a satisfactory result on the bigger dataset MSR-VTT does not indicate that, even though sufficient semantics are involved in the proposed framework, further fine-tuning of parameters may result in an improved performance. Another experimental finding is that METEOR is more favoured and can also capture semantic characteristics. Table 2 is the experimental evidence for the METEOR performance.

As a part of ablation study we performed several other experiments. The outcomes of our suggested framework with dropout parameters [47] are reported in Table 4 and indicates that the framework discovers more value when nodes are not dropped than when nodes are dropped. This could be because our two-layer attention strategy is connected with layered bidirectional encoders and unidirectional decoders. Due to the framework's two-layered attention, it intelligently selects the best fragments or visual frames from which to learn and anticipate new values without interfering with the growth or reduction of network nodes.

In addition to VGG-16, the proposed system is evaluated using another feature vector named NASNet Feature Extractor [48]. Table 5 summarises the findings. As demonstrated in Table 5, this Feature Extractor explored with and without a drop

Table 5 Performance score with different Feature Extractor

MODEL	MSVD				
	B@1	B@2	B@3	B@4	METEOR
Multi-layer attention (Proposed) without dropout and NASNet Feature Extractor	60.10	41.27	34.36	19.81	39.40
Multi-layer attention (Proposed) with dropout and NASNet Feature Extractor	58.29	38.87	31.65	17.16	42.37

layer. Our model fared better when no drop out is included in the BLUE performance metric, however METEOR performs better when a drop layer and NASNet feature extractor are included. This illustrates that, as previously demonstrated, the multi-attention framework optimises to select the best segments or frames, resulting in a high METEOR score.

Limitations

The experimental studies revealed that the proposed system performed admirably on the training part of the standard dataset used. However, the outcome on the test portion is less than the result on the training portion. Though the framework earned a higher METEOR score than any previous study, it fell short of achieving a superior BLUE score. Whereas appropriate measures must be taken to increase the BLEU score. The suggested structure takes into account the average length of phrases used during training. As a result, it may exclude some critical terms from the sentences. Additional study in this area may help improve the model's performance.

Conclusions

It is proposed in this research to use a novel Multi-Layer Attention-based approach for video captioning that is both efficient and effective, and it is then compared to other modifications of the base model. The proposed framework consists of a visual encoder and a language model, which are primarily configured with LSTM networks.

The visual encoder is implemented using stacked Bi-LSTMs on resampled video data in order to maintain input at every time interval for attention. The encoder's hidden states are then used to create a global perspective of the video, which is subsumed into the language model. The decoder unit, which was utilised to convert the video captions into sentences word for word. The framework's implementation was carried out on the MSVD and MSR-VTT datasets, respectively. To the best of our knowledge, the suggested approach surpassed practically most of the existing state-of-the-art visual captioning results that have been published and listed. The best way to increase the effect in the future is to modify it even further in order to achieve the best outcome on a larger dataset.

Acknowledgements
Not applicable.

Author contributions
Both authors contributed to design and implementation, analysis of results, and preparation of the manuscript. Both authors read and approved the final manuscript.

Availability of data and materials
Not applicable.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 5 February 2022 Accepted: 20 October 2022
Published online: 12 November 2022

References

1. Shorten C, Khoshgoftaar TM, Furht B. Text data augmentation for deep learning. J Big Data. 2021;8(1):1–34.
2. Aneja J, Deshpande A, Schwing A. Convolutional image captioning. Comput Vis Pattern Recognit. 2017;1711:09151.
3. Kiros R, Salakhutdinov R, Zemel RS. Unifying visual-semantic embeddings with multimodal neural language models. https://arxiv.org/abs/1411.2539
4. 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. 2017. p. 706–715.
5. Amirian S, Rasheed K, Taha TR, Arabnia HR. Automatic image and video caption generation with deep learning: a concise review and algorithmic overlap. IEEE Access. 2020;8:218386–400.
6. Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, Santamaría J, Fadhel MA, Al-Amidie M, Farhan L. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. J Big Data. 2021;8(1):1–74.
7. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9(8):1735–80.
8. Bin Y, Yang Y, Shen F, Xie N, Shen HT, Li X. Describing video with attention-based bidirectional lstm. IEEE Transact Cybern. 2018;49(7):2631–41.
9. Li S, Tao Z, Li K, Fu Y. Visual to text: survey of image and video captioning. IEEE Transact Emerg Topics Comput Intell. 2019;3(4):297–312. https://doi.org/10.1109/TETCI.2019.2892755.
10. Yang Y, Zhou J, Ai J, Bin Y, Hanjalic A, Shen HT, Ji Y. Video captioning by adversarial lstm. IEEE Transact Image Process. 2018;27(11):5600–11. https://doi.org/10.1109/TIP.2018.2855422.
11. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint. 2014. https://arxiv.org/abs/1409.1556
12. Krishnamoorthy N, Malkarnenkar G, Mooney R, Saenko K, Guadarrama S. Generating natural-language video descriptions using text-mined knowledge. In: Twenty-Seventh AAAI Conference on Artificial Intelligence. 2013.
13. 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. 2016. p. 1029–1038.
14. 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 COLING 2014, the 25th International conference on computational linguistics: technical papers. 2014. p. 1218–1227.
15. Xu R, Xiong C, Chen W, Corso J. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In: Proceedings of the AAAI conference on artificial intelligence. 2015; 29.
16. Jaafari J, Douzi S, Douzi K, Hssina B. Towards more efficient cnn-based surgical tools classification using transfer learning. J Big Data. 2021;8(1):1–15.
17. 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. 2015. p. 4534–4542.
18. 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. 2015. p. 4507–4515.
19. Bin Y, Yang Y, Shen F, Xu X, Shen HT. Bidirectional long-short term memory for video description. In: Proceedings of the 24th ACM International conference on multimedia. 2016. p. 436–440.
20. Gao L, Guo Z, Zhang H, Xu X, Shen HT. Video captioning with attention-based lstm and semantic consistency. IEEE Transact Multimed. 2017;19(9):2045–55.
21. Zheng Y, Zhang Y, Feng R, Zhang T, Fan W. Stacked multimodal attention network for context-aware video captioning. IEEE Transact Circuit Syst Video Technol. 2022;32(1):31–42. https://doi.org/10.1109/TCSVT.2021.3058626.
22. Deng J, Li L, Zhang B, Wang S, Zha Z, Huang Q. Syntax-guided hierarchical attention network for video captioning. IEEE Transact Circuit Syst Video Technol. 2022;32(2):880–92. https://doi.org/10.1109/TCSVT.2021.3063423.
23. Hua X, Wang X, Rui T, Shao F, Wang D. Adversarial reinforcement learning with object-scene relational graph for video captioning. IEEE Transact Image Process. 2022;31:2004–16.
24. Venugopalan S, Xu H, Donahue J, Rohrbach M, Mooney R, Saenko K. Translating videos to natural language using deep recurrent neural networks. 2014. . http://arxiv.org/abs/1412.4729
25. Zhao B, Li X, Lu X. Cam-rnn: co-attention model based rnn for video captioning. IEEE Transact Image Process. 2019;28(11):5552–65. https://doi.org/10.1109/TIP.2019.2916757.

26. Gao L, Li X, Song J, Shen HT. Hierarchical lstms with adaptive attention for visual captioning. IEEE Transact Pattern Anal Mach Intell. 2020;42(5):1112–31. https://doi.org/10.1109/TPAMI.2019.2894139.
27. Hossain MZ, Sohel F, Shiratuddin MF, Laga H, Bennamoun, M. Bi-san-cap: bi-directional self-attention for image captioning. In: 2019 Digital image computing: techniques and applications (DICTA). 2019. p. 1–7. https://doi.org/10.1109/DICTA47822.2019.8946003
28. Xu J, Yao T, Zhang Y, Mei T. Learning multimodal attention lstm networks for video captioning. In: Proceedings of the 25th ACM International conference on multimedia. MM '17. New York: Association for computing machinery; 2017. p. 537–545. https://doi.org/10.1145/3123266.3123448
29. Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. In: Advances in neural information processing systems. 2014. p. 3104–3112.
30. Schuster M, Paliwal KK. Bidirectional recurrent neural networks. IEEE Transact Signal Process. 1997;45(11):2673–81.
31. Karpathy A, Fei-Fei L. Deep visual-semantic alignments for generating image descriptions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015. p. 3128–3137.
32. Ullah A, Ahmad J, Muhammad K, Sajjad M, Baik SW. Action recognition in video sequences using deep bi-directional lstm with cnn features. IEEE Access. 2017;6:1155–66.
33. Li J, Qiu H. Comparing attention-based neural architectures for video captioning. https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194
34. 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: International conference on machine learning. New York: PMLR; 2015. p. 2048–2057.
35. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. 2014. arXiv. http://arxiv.org/abs/1409.0473
36. Chen D, Dolan W. Collecting highly parallel data for paraphrase evaluation. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies. Portland: Association for Computational Linguistics. 2011. p. 190–200.
37. Xu J, Mei T, Yao T, Rui Y. Msr-vtt: A large video description dataset for bridging video and language. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 5288–5296 (2016)
38. Zeiler MD. ADADELTA: an adaptive learning rate method. CoRR. 2012. https://arxiv.org/abs/1212.5701
39. Papineni K, Roukos S, Ward T, Zhu WJ. Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the association for computational linguistics. 2002; p. 311–318 (2002). https://doi.org/10.3115/1073083.1073135
40. 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: Association for Computational Linguistics. 2005. p. 65–72.
41. Feinerer I, Hornik K. Wordnet: WordNet Interface. R package version 0.1-15. 2020. https://CRAN.R-project.org/package=wordnet
42. Pan Y, Mei T, Yao T, Li H, Rui Y. Jointly modeling embedding and translation to bridge video and language. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 4594–4602.
43. Xu H, Venugopalan S, Ramanishka V, Rohrbach M, Saenko K. A multi-scale multiple instance video description network. 2015. . http://arxiv.org/abs/1505.05914
44. Venugopalan S, Hendricks LA, Mooney R, Saenko K. Improving lstm-based video description with linguistic knowledge mined from text. . http://arxiv.org/abs/1604.01729
45. Yan C, Tu Y, Wang X, Zhang Y, Hao X, Zhang Y, Dai Q. Stat: Spatial-temporal attention mechanism for video captioning. IEEE Transact Multimed. 2019;22(1):229–41.
46. Aafaq N, Akhtar N, Liu W, Gilani SZ, Mian A. Spatio-temporal dynamics and semantic attribute enriched visual encoding for video captioning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2016. p. 12487–12496.
47. 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(1):1929–58.
48. Zoph B, Vasudevan V, Shlens J, Le QV. Learning transferable architectures for scalable image recognition. InProceedings of the IEEE conference on computer vision and pattern recognition. 2018. p. 8697–8710.

Publisher's Note