Video Captioning with Reconstruction Network

Durga Yasasvi **IMT2016060**
Siddarth Reddy **IMT2016037**
Srujan Swaroop **IMT2016033**
Rohith Yogi **IMT2016072**

Abstract:
Here we present a a simple, flexible and scalable approach to rediscover the idea of reconstruction network for Video captioning along with exploiting the functionalities of LSTM's and attention mechanism driven by Encoder-Decoder architecture.Unlike the general approaches that works on the principle of forward flow using Encoder-Decoder that generates caption descriptions in language based on the latent representations of the video content from the encoder ,here in this approach we employ reconstruction network on top of Encoder-Decoder network to promote backward flow that reconstruct video frame representations from the hidden state sequence of the language decoders.The generation loss obtained by the encoder-decoder part and the reconstruction loss that had been introduced by the reconstruction network part are combined and drawn into training the reconstruction network in an end-to-end fashion.This essentially trains the model in such a way that more and more information is captured in hidden state representations which are used for generating captions.Also each component in the architecture is customized with the motive to make them simple,convenient,efficient and compatible.The customization is done in accordance with the literature reference which will be in detail discussed in the report.

1. Introduction

In the context Video captioning is the term framed to refer the task of generating natural language sequences which describes the visual contents of the video. The idea of describing visual contents of a video in a natural language seems very fascinating but at the same time very complex and sophisticated. As it requires a consistent framework to exploit it's all external and internal dimensions. Some of the major attributes and factors we have figured out for the task of video captioning is as follows:-

- Capturing Salient content like objects,scene,background.
- Semantic content like actions or activities,subjects.
- Learning Contextual knowledge of the event and environment of the video.
- Understanding dynamic and static relationships in the context(i.e spatial dynamics).
- And the most important is temporal dynamics of the video.

Now we can understand the underlying complexities in achieving the task of video captioning as modelling all these diverse range of factors and capturing such intense information is definitely not an easy task.

In order to propose a simple, consistent and working framework to solve the problem of video captioning in an efficient manner also in such a way that it captures all the above mentioned necessary attributes for the task, we have referred all the literature present in the context of video captioning to develop a deeper understanding and to come with a proper model which not only just acts as a manifestation of best possible approach but also as a solution to anomalies of the previous approaches.

So with all understanding and analysis the best possible approach for the task of video captioning we could arrive at is **Video captioning with Reconstruction Network**.

2. Idea of Reconstruction Network for Video Captioning

As mentioned in the above section we have decided to manifest the approach based framework called **Video captioning with Reconstruction Network**. In this approach a special type of network called reconstruction network is introduced. Here along with video to caption sentence(-forward flow) transformation, reconstruction network imposes an additional workflow to generate video from sentences(-backward flow). This presents just a high level overview of the model it can be more granularly presented as an idea of dual flows as follows:-

- Forward flow(Video to sentence) :-This essentially means to generate the natural language captions for the video based on the encoded semantic contextual features by the **encoder decoder network**.
- Backward flow(Sentence to video) ;- The backward flow from sentence to video is meant to reproduce the video representation features from the hidden state sequence of the decoder by the **reconstruction network**.

So the essential purpose of the reconstruction network is to enforce the encoder-decoder model to learn features to incorporate more and more information of the input video sequence. The ultimate claim which is purposely made is that if the hidden state sequence generated by the decoder has enough information to reproduce the video frame representation features as close as it can through reconstruction then the semantic and contextual gap between the natural language caption sequences and input video sequences is essentially said to be bridged, which is the eternal realization and the motive of the task of video captioning that we are meant to perform according to the essence of the literature that we have referred.

The analysis and understanding of the literature which we have referred for the corresponding study is mentioned in detail in subsequent sections.

3. Learning through Literature

As discussed to develop a deeper understanding and to get the sense of the intrinsic details of the video captioning task we have referred a bunch of literature present in the context. Learning from literature not only helps us getting intrinsic insights of their respective design and implementation aspects but also gets us the glimpse of diversity of approaches designed for the same problem of video captioning. This helps us to approach problem not in a single direction of thought but to think in multiple perspectives and angles of approach.

Before analyzing literature we would like to present some common technical terminologies from the literature in the context of video captioning. In understanding semantics of the video frame features two concepts of features arise as follows:-

- Static Semantic features which refers to the object,person, scene, background in the video frames.
- Dynamic Semantic features corresponds to the activity or action performed in the video.

In reference to the natural language descriptions Static semantic features corresponds to the nouns in the caption sentences and Dynamic semantic features refer to the verbs in the caption sequences. The essence of the above mentioned aspects is to state that for capturing the information in video besides embedding just the visual features of the video the approach should essentially embed the spatial and temporal dynamics along with contextual knowledge which

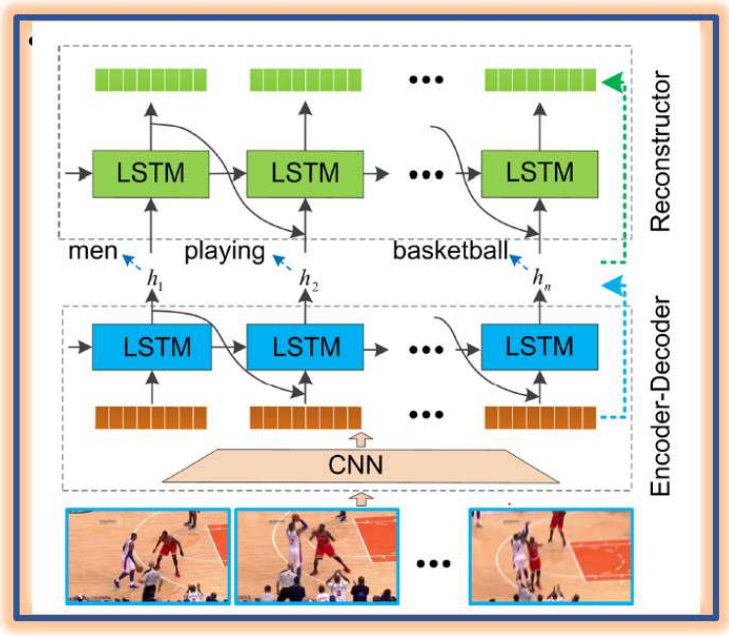

Fig. 1. Architecture of Reconstruction network for Video captioning.

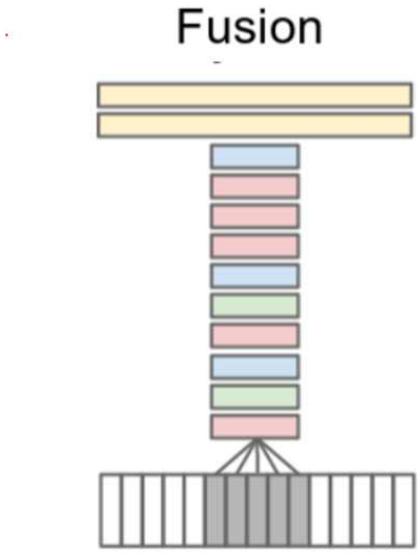

Fig. 2. Fusion of several streams to make final decision

altogether comprised only can completely describe the contents of the video(Subjects ,objects ,activity ,context etc.).
The literature which we are about to discuss incorporates these features in some or the other way. So as said we are going to study and analyze the way they manifested these features and technical complexities they faced and the work-around designed by them to solve the complexities with the advancement of time and thinking.

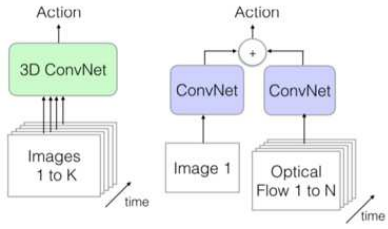

Fig. 3. Optical stream is added to model temporal aspects of the video

Our exploration of literature started from the learning about Action and Gesture recognition as it is a similar kind of task.The development has happened through many stages with the time. This development is briefly mentioned here:-

- To capture the visual and spatial features of the video, video is fragmented into several frames and individual frames at different streams with respect to time are treated with 2D convolutional network separately and finally all streams are fused together to make final decision. The fusion used to occur in different formats like Early fusion, Slow fusion, Late fusion etc.
- But in the above approach contextual dimension and capturing high level picture of the video was absent , soon realizing this they came with novel approach of extracting 3D features of the video using 3D convolutional network incorporating time dimension to present contextual and high level representation of the video.
- Also an additional stream called context stream consisting of low resolution images is added along with the regular streams to capture the contextual environment of the video.
- Last but not least to model the temporal aspects of the activities optical flow between consecutive frames are provided as input to the model along with the sampled frames of the video.

But there are some fundamental differences between Activity Gesture Recognition and Video Captioning which can be categorized as follows:-

- The time duration of the video or number of frames in the video considered for the task is relatively very large.
- In Gesture/Activity recognition by considering all frames it need not generate caption sequence it just have to classify the activity or gesture one among several predefined classes. But in video captioning we need to create an explicit language model for caption generation.

Apart from modelling temporal and spatial aspects along with preserving semantic structure the model should also pay attention to the above mentioned aspects(i.e handling longer duration of frames and creating language model for caption generation). Several models have experimented with different variations to explore these aspects of the model. We are going to very briefly detail them in concrete manner along with the key understanding we have acquired from them.

In initial methods of video captioning for capturing semantic, temporal and contextual content included sequential processing of multiple frames of video with time without considering temporal dependencies. Once the required information is retrieved from the video they need to systematically arrange linguistically in the form of a natural language caption. For this they classify dynamic semantic content and static semantic content into verbs and nouns respectively. And with the help of contextual information they identify the subject, candidate objects ,scene etc. Using this information they generate a sentence based on a template, they form a template based language model modelled by a probabilistic graphical model(PGM) to put the sequence of words in an order

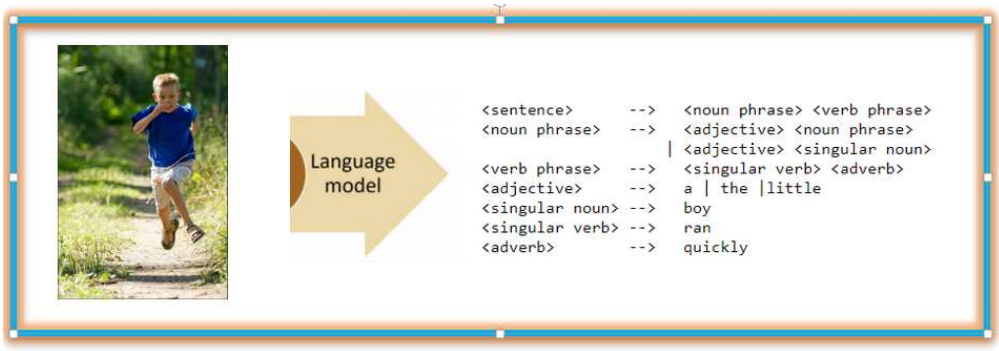

Fig. 4. A sample illustration of Captioning using Language template.

based on their respective confidence scores.

One thing that should be noted here is that this is a two stage pipeline:-

- Retrieving the appropriate and required content from the video frames.
- Generating language model based on the content retrieved in the first stage.

One major drawback in this type of architecture is that the template based language model cannot be scaled. The sentences are constrained to a fixed format and structure.

So the community decided to implement a flexible and non-rigid language model to generate caption sequences such that there is no communication gap in conveying the content extracted (from videos) linguistically in the form of meaningful and sensible language sentences without any constraints and rigidity.

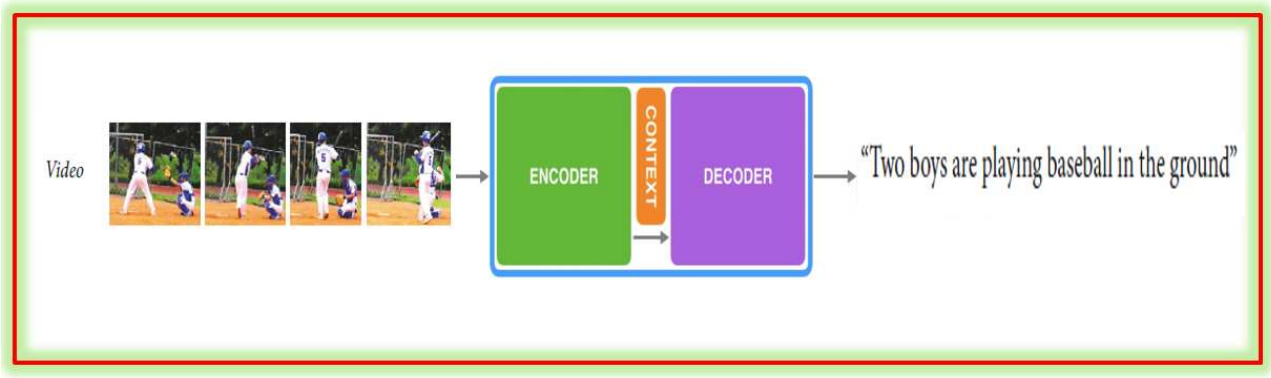

Fig. 5. An Encoder-Decoder model for video captioning without attention.

In order to build a language model it's temporal dependencies and relationships must also be modelled as each word generated in a sentence is very much dependent on it's neighbouring words and also the context.So with the advent of Recurrent Neural Networks(RNN's) in the field of machine translation,speech recognition they are also introduced in the field of vision in the tasks of image captioning and video captioning.
But in particular coming to the area of video captioning the RNN need to take the encoded representations of video frames in the form of context input and decode the caption sentence word by word. Hence this set of architecture is termed as *Decoder* Model.
In similar context in initial stage of the model the video frames are processes sequentially with

RNN's or CNN's to produce encoded features which encodes enough information(like subject,object,scene ,context,background etc) and feed as context input to the decoder,hence this phase of architecture is called *Decoder Model*.
These two models work together in co-ordination to achieve the task of video captioning and compositely called as **Encoder-Decoder model**.
Each of them has a corresponding functionalities to perform as detailed below:-

- **Encoder**:- Encoder takes the frames of the video in the temporal fashion and encodes them to produce encoded semantic representations of them at different time steps and provides them to decoder for generating caption sentences.
- **Decoder**:- Decoder receives the encoded frame representations from the decoder and try to decode the corresponding captions word by word. Hence decoder is trained to act as language model conditioned on the input feature representations of the video frames from the encoder and generate captions according to it.

But one major problem in sequential decoding using RNN is that RNN's are very ineffective in learning long term temporal dependencies as a solution to this RNN's are replaced with LSTM's(Long Short Term Memory) by introducing cell states and regulating the flow of information by the gates.
But soon it was realized the encoding the entire video into single state and passing it into decoder as context is not completely justifiable as the particular decoding steps might require information about only particular hidden states of the encoder, hence this problem is solved by the method called attention mechanism where, instead of only encoding the entire video frame representations, so the mechanism learns to weigh the frame features non uniformly according to their relevance to the corresponding decoding stage rather than uniformly weighing representations of the frames from the encoder.

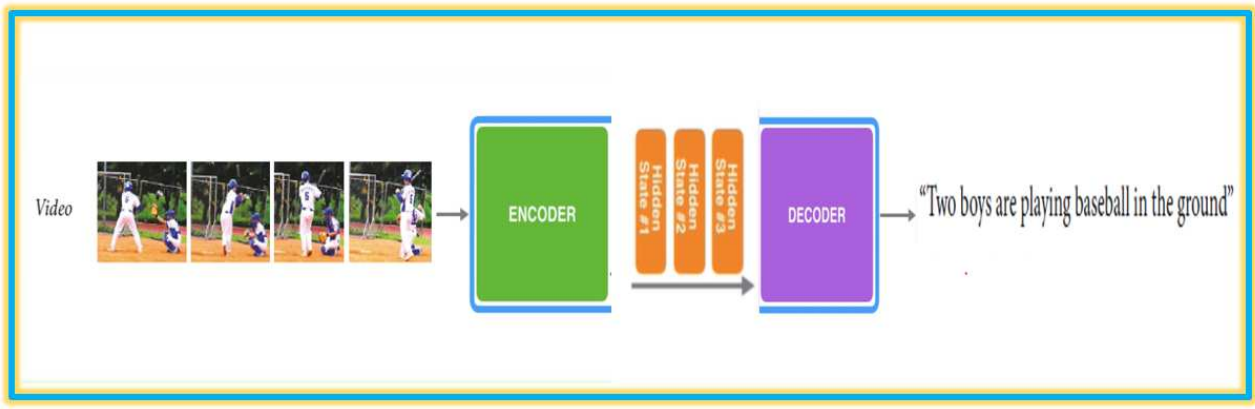

Fig. 6. An Encoder-Decoder model for video captioning with attention

Currently in the Computer Vision community the standard model for Video captioning is the Encoder-Decoder model with LSTM as sequence encoders(for encoding video frame representations) and decoders (for decoding caption sequences conditioned on hidden state frame representations from encoder) enabled with attention mechanism between Encoder stage and Decoder stage.
Through the course of studying literature we have drawn some critical conclusions , acquired some learning ,identified the key aspects required for the task and some crucial areas where attention should be paid. So we have listed our key understandings and the purpose for selecting

the current paradigm of **reconstruction network** in the following section.

4. Purpose of Reconstruction Network for Video Captioning

After analyzing and understanding the literature we have identified some key takeaways in the form of problems and solutions that should be note before proceeding to accomplish the task of video captioning as listed below:-

- ***Problem***:-The model should incorporate a way to distinctively capture static and dynamic semantic visual features and able to implicitly distinguish them in the architecture.
  Solution:-This is one of the primary basic prerequisite for video captioning, this can be resolved by finding representation feature vectors of video frames by passing them to already trained convolutional networks (pre-trained networks).
- ***Problem***:-The model should also learn spatial and temporal dynamics in and across frames of the video.
  Solution:-This can be accomplished by employing LSTM's in encoders as they can manage temporal dependencies across the encoded frames representations and pass it to decoder.
- ***Problem***:-While decoding the encoded features the decoder should retrieve the required and relevant context from the encoder .
  Solution:-This can be solved by inducing attention mechanism between encoder stage and decoder stage as the weightage is distributed among encoded features according to the relevance or attention the decoder should pay at that particular time step.
- ***Problem***:-Most important the model must have an explicit language model to generate natural language caption sequences corresponding to the video.
  Solution:-The problem to this solution lies in the encoder-decoder type architecture itself, the decoder itself acts as language model as it generates captions word by word conditioned on the frame semantic representation features.

  But in every model that we have analyzed ignores the huge semantic gap between high level video frame feature representations and low level natural language sentence descriptions since the flow is one direction -forward flow i.e the models are designed always to flow from videos to sentences. The hidden states in decoder which are used to generate caption descriptions are regulated by weights which in turn are updated according to gradient loss propagated. But never ever there is mechanism employed to ensure whether hidden state sequence generated by decoder has enough information to decode and generate captions.
  This is where we bring in the idea of Reconstruction Network where we induce a method to reproduce video frame representations from hidden state sequences of the decoder. So if we train the model such that video frame representations are reproduced from decoder hidden state sequence then we can subjectively claim that hidden states of decoder posses enough latent semantic information to produce the corresponding captions for the video.
  Hence by incorporating this bi-directional flow of learning i.e from video-to-sentences and sentences-to-video we are ensuring and enforcing that more information of video is embedded into the decoder hidden states. Thereby enhancing the relationship between the generated captions and video frame representations which assists in reducing the semantic gap between the Video frame representations and generated caption sentences.
  So along with the discussed key attributes we incorporate this idea of reconstruction network and present a composite model which is collection of key architectural ideas inspired from the literature and this reconstruction network. The complete model architecture, design and the modifications will be discussed in the subsequent sections.

So here we propose a paradigm for instance segmentation, which by design also performs object detection and classification. Instance segmentation is definitely not a very easy problem to tackle when compared to actions like object detection or classification, because this it in turn

demands some of these actions to be performed as intermediate steps in the process. Some of them are :-

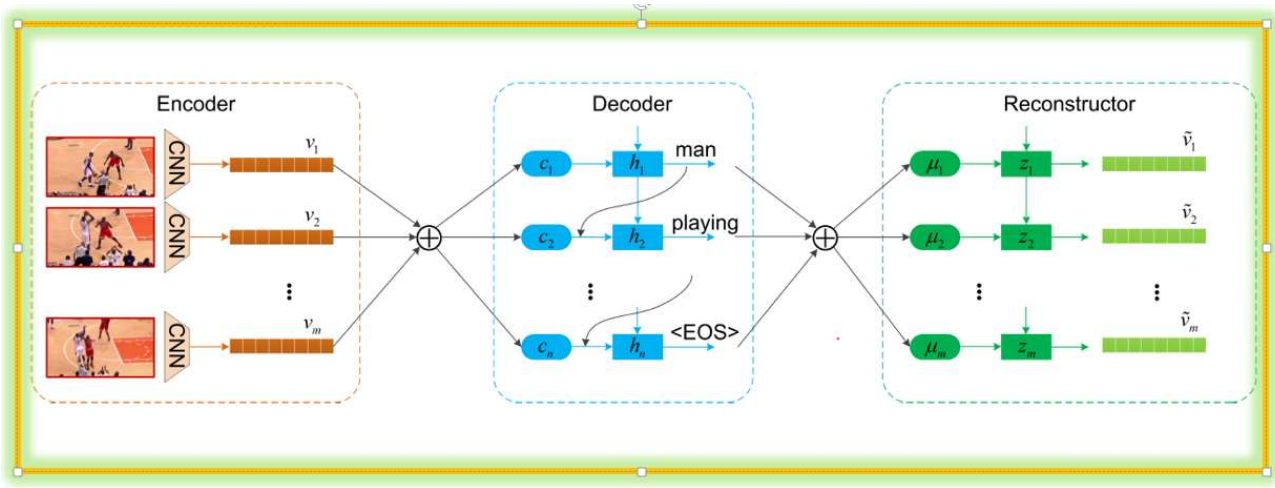

Fig. 7. Underlying model of encoder-decoder-reconstruction network before attributing the change

5. Model

It is a Reconstruction Network with an Encoder-Decoder-Reconstruction architecture, that holds both the forward flow (video to captions) and the backward flow (captions to video). The Encoder-Decoder is responsible for the forward flow to produce captions from the video input. And the Reconstruction is responsible for the backward flow that reproduces the video representation from the hidden state sequence generated by the Decoder. We customize two types for reconstruction, one that focuses global structures and the other local structure of the video sequence.

In general the encoder is a simple CNN that captures the image structure and produces its semantic representation. All the video frame's representation are blend together to utilize the video's temporal dynamics and produce the video representation. Whereas the Decoder usually contains LSTM's stacked upon each other. Basically the Encoder-Decoder generates the semantic representation of each video frame and later that generates a sentence description or a caption.

The dual learning approach which was famously used for Neural Machine Translation was inspired for the Reconstruction part, where the source sentences are reproduced from the target hidden states. As the reconstruction urges a constraint that the semantic information of a video is reconstructed from the hidden state sequence yielded by the Decoder, hence the Encoder-Decoder are encouraged to incorporate more semantic information about the source video.

5.1. Encoder-Decoder

This part of the model aim is to generate a caption $S = \{s_i, s_2, s_3, ..., s_n\}$ for the video V. So, the captioning probability generation word by word is:

$$P(S|V) = \prod_{i=1}^{n} P(s_i|s_1, s_2, ..., s_n, V; \theta)$$

where θ is set of parameters of the encoder-decoder. n denotes the length of the sequence and $s_1, s_2, s_3, ..., s_(i-1)$ denotes the partial caption generated.

We advocate using Inception-V4 architecture into the Encoder. Given a video input the encoder, it is encoded as a video sequence representation $V = v_1, v_2, ..., v_m$ where m denotes the total number of frames. And the decoder aims to yield captions word by word based on the video

representation. In order to utilize the global temporal information, a temporal attention mechanism is used to make the decoder to select the main/key frames for captioning:

$$P(s_i|s_<i, V; \theta) \propto exp(f(s_{i-1}, h_i, c_i; \theta))$$

here, f represents the LSTM activation function, h_i is the i_th hidden state and c_i denotes the i_th context vector computed using the temporal attention mechanism. The temporal attention mechanism is used to assign weight α_i^t to the hidden states of the encoder $\{h_1, h_2, ..., h_i, ...\}$ at the time step t as follows:

$$c_t = \sum_{i=1}^{m} \alpha_i^t h_i$$

where m denotes the number of video frames, α_i^t reflects the relevance of the i_th temporal feature in the video sequence given all previously generated words. The encoder-decoder model is trained minimizing the negative log likelihood as follows:

$$min_\theta \sum_{i=1}^{N} \{-\log P(S_i|V_i; \theta)\}$$

Coming to the reconstruction network, it aims at re-generating the sequential video frame representations produced by the encoder, with the hidden states $H = h_1, h_2, ...h_n$ of the decoder as input. Two types of reconstruction networks are customized one reproducing the global structure and the other focuses at reproducing the local structure using self attention like model).

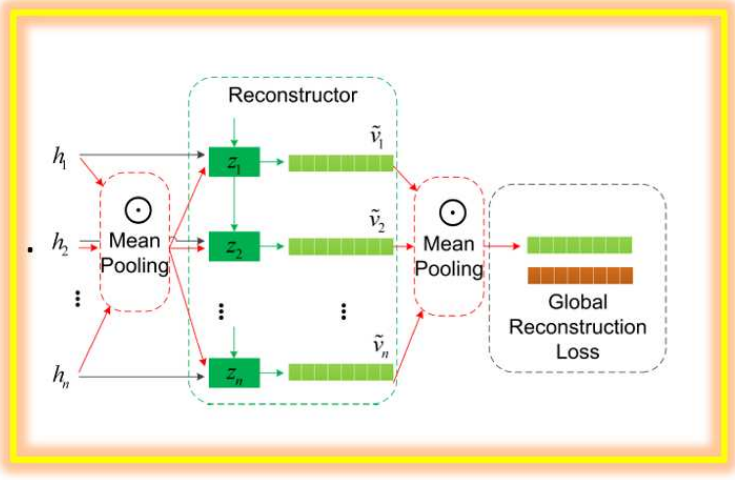

Fig. 8. Demonstration of the reconstruction network reproducing the global structure

5.2. Reconstruction-Global Structure

The complete sentence is considered along with the hidden states h_t that characterizes whole semantics of the sentence. In order to produce the global representations of the caption, a mean pooling strategy is performed on the hidden states of the decoder:

$$\phi(H) = \frac{1}{n} \sum_{i=1}^{n} h_i$$

where $\phi(.)$ denotes the mean pooling process that yields a vector representation $\phi(H)$. The LSTM is identified as:

$$\begin{pmatrix} i_t \\ f_t \\ o_t \\ g_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{pmatrix} T \begin{pmatrix} h_t \\ z_{t-1} \\ \phi(H) \end{pmatrix}$$

$$m_t = f_t \odot m_{t-1} + i_t \odot g_t$$

$$z_t = o_t \odot tanh(m_t)$$

where i_t, f_t, m_t, o_t and z_t denote the input, forget, memory, output and hidden states of each LSTM unit, respectively. σ and $\odot$ denote the logistic sigmoid activation and the element-wise multiplication, respectively. And the reconstruction loss is defined as:

$$L^g_{rec} = \psi(\phi(V), \phi(Z))$$

where $\phi(Z)$ denotes the ground truth of the video sequence generated by the mean pooling process, whereas $\phi(V)$ denotes for the frame features. And $\psi(.)$ denotes the Euclidean distance.

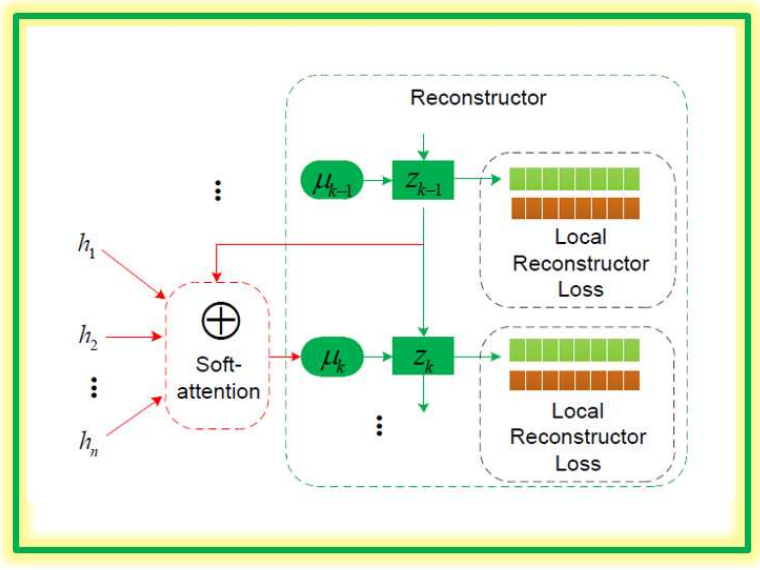

Fig. 9. Illustration of the reconstruction network reproducing the local structure

5.3. Reconstruction-Local Structure

Unlike in the global reconstruction network, we preserve the local temporal dynamics by reconstructing each video frame. Basing the self attention strategy, from the key hidden states of the decoder we reproduce the feature representation of each frame:

$$\mu_t = \sum_{i=1}^{n} \beta_i^t h_i$$

where $\sum_{i=1}^{n} \beta_i^t = 1$ and β_i^t denotes the weight computed for the $i_t h$ hidden state in the caption given all previously computed frame representations $\{z_1, z_2, ..., z_{t-1}\}$. This forces the reconstruction network to work on the hidden states by adjusting the attention weight β_i^t and yields the

context information μ_t. The LSTM is identified as:

$$\begin{pmatrix} i_t \\ f_t \\ o_t \\ g_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{pmatrix} T \begin{pmatrix} \mu_t \\ z_{t-1} \end{pmatrix}$$

And the reconstruction loss is defined as follows:

$$L_{rec}^{g} = \frac{1}{m} \sum_{j=1}^{m} \psi(z_j, v_j)$$

5.4. Training Overview

The complete training proceeds in two stages,which we will be discussed in subsequent sections in detail. first we depend on the forward likelihood to train the encoder-decoder component. Later, we rely on the backward reconstruction loss $L_{rec}\theta_{rec}$. The reconstruction loss is calculated using the hidden state sequence generated by the LSTM units in the reconstruction network as well as the video frame feature sequence.

6. Experimental Setup

We have worked on two different datasets namely MSRVTT AND MSVD.

6.1. MSR VTT

In its current version, MSR-VTT provides 10K web video clips with 41.2 hours and 200K clip-sentence pairs in total, covering the most comprehensive categories and diverse visual content, and representing the largest dataset in terms of sentence and vocabulary. Each clip is annotated with about 20 natural sentences.

6.2. MSVD

This a dataset of youtube clips which contains about 1970 videos with the corresponding annotations provided. Average length of each video is about 10 seconds which corresponds to about 330 minutes for total dataset.

6.3. Dividing the dataset

In MSR-VTT dataset, we only worked on two different categories namely Movies and sports. First we considered only sports category which consisted of 750 videos each about of 20 seconds.Second, we considered movies category assuming describing movies would be more diffucult and wanted to check how our model works on them. Movies category consists of about 700 images each about a length of 20 seconds. We split these images into train, validation and test images.

We extracted the features of images of sports/movies category numbered 3/7 in dataset by reading the hdf5 file which consists of features of all the images of MSR-VTT dataset. We then sent these extracted features to the encoder-decoder model which generates the captions and these captions are sent to reconstruction layer where we reconstruct the frames(images) features from the obtained captions. To encapsulate more accurate details of videos in captions we use reconstruction layer. We can implement this by optimizing the loss function which minimizes the euclidean between reconstructed features and original features throughout the training process.

For MSVD dataset we worked on the whole dataset. Split the whole dataset into train, validation, test images and extracted features of these images. Followed the same process described above to obtain captions.

7. Training

Basically our architecture consists of two parts which we trained separately in two stages namely:

- **Encoder-Decoder – training stage 1**
- **Encoder-Decoder-Reconstructor – training stage 2**

7.1. Encoder-Decoder

In this stage the frames which are sent as input to this architecture are encoded using the encoder part which captures the high level semantic information about the video as features. These features are sent to the decoder architecture which consists of LSTM's generates captions.

7.1.1. Loss function

The encoder-decoder model can be jointly trained by minimizing the negative log likelihood to produce the correct description sentence given the video as follows:

$$min_\theta \Sigma_{i=1}^{N}(-logP(S^i|V^i;\theta))$$

Here each V_i means a video which is represented as m frames $(v_1, ..., v_m)$

7.2. Encoder-Decoder-Reconstructor

Reconstructor is trained on the top of the encoder-decoder in the second stage, which is expected to reproduce the video from the hidden state sequence of the decoder. However, due to the diversity and high dimension of the video frames, directly reconstructing the video frames seems to be impossible. Therefore, the reconstructor aims at reproducing the sequential video frame representations generated by the encoder, with the hidden states H = $(h1, h2, ..., hn)$ of the decoder as input .

7.2.1. Loss function

The reconstruction loss is just the euclidean distance between reconstructed hidden states and encodings generated by encoder network.

$$d(u, v) = \sqrt{\Sigma_{i=1}^{N}(u_i - v_i)^2}$$

$$L_{rec_i} = \Sigma_{i=1}^{m}(d(u, v))$$

This is for one video which has m frames. We can add these over all videos in the dataset.So, the loss function for this stage is:

$$L(\theta, \theta_{rec}) = \sum_{i=1}^{N}(-\log P(S_i|V_i;\theta) + \lambda L_{rec}(V_i, Z_i;\theta_{rec})$$

7.3. Training parameters

MSR-VTT dataset:
The model was trained for about 30 epochs during **stage-1** where the batch size consists of 200 videos. Initially, while training the learning rate is $2e-4$ and the weights were optimized using Adam optimizer. For **stage-2**, the model was trained 30 epochs and the learning rate is $4e-5$ with same batch size and optimizer.
MSVD dataset:
The model was trained for about 50 epochs during **stage-1** where the batch size consists of 200 videos. Initially, while training the learning rate is $5e-5$ and the weights were optimized using Adam optimizer. For **stage-2**, the model was trained 30 epochs and the learning rate is $2e-5$ with same batch size and optimizer.

8. Evaluation Metrics

Describing video content precisely is difficult because videos are visually rich and because their content can be interpreted in so many ways. So, to assess the captions we obtain from our model we followed the most popular metrics for the quantitative evaluation of the model:

1) **METEOR**
2) **CIDEr**
3) **BLEU**

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in \{Candidates\}} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')}.$$

Fig. 10. Evaluated on different metrics

Since our primary metric is BLEU we would like to explain the working methodology of BLUE metric.
BLEU (Bilingual Evaluation Evaluation Understudy) approach works by counting the matching n-grams in the candidate translation to n-grams in the reference text. Similarly, METEOR uses only uni-gram method for counting matches but uses stemming and synonyms for better evaluation. Also METEOR instead of looking only at precision, it also considers Recall into the account.

Consider the below example for Unigrams.

- French: Le Chien est sur le sofa.
- Human Ref 1: The dog is on the sofa.
- Human Ref 2: There is a dog on the sofa.
- Machine Translation: The the the the the the the.

Precision on uni-grams(P_1): 2/7, Uni-grams are individual words of a sentence.
Numerator: max(no. of 'the' words in ref1, no. of 'the' words in ref2),
Denominator: no.of 'the' words in machine translation.
BLEU on Bi-grams:

- French: Le Chien est sur le sofa.
- Human Ref 1: The dog is on the sofa.
- Human Ref 2: There is a dog on the sofa.
- Machine Translation: The dog the dog on the sofa.

Bi-grams are the pairs of words appearing next to each other. Possible Bi-grams of Machine Translation output are (the dog), (dog the), (dog, on), (on the), (the sofa).

Bi-grams	Count in Machine Translation	Count in Ref 1 or 2 (Max count in either of the Ref 1 or 2 is taken)
(the dog)	2	1
(dog the)	1	0
(dog on)	1	1
(on the)	1	1
(the sofa)	1	1

Precision on Bi-grams(P_2): 4/6.

Numerator: total no. of bi-grams matching w.r.t. the references.
Denominator: total no. of bi-grams.
Precision of N-grams(P_n):

$$P_n = \frac{\sum_{n-grams \epsilon \hat{y}} Count_{ref}(n-grams)}{\sum_{n-grams \epsilon \hat{y}} Count_{MT}(n-grams)}$$

BLEU Allows you to measure the degree to which the machine translation output is similar to the human references. If the MT output is exactly similar to the references, then the precision $P1, P2, \ldots Pn$ will be equal to 1. Combined BLEU score:

$$BLEU = BP * exp(\sum_{n=1}^{N} w_n log(P_n))$$

$$BP : BrevityPenalty = \begin{cases} 1 & if c > r \\ e^{1-\frac{r}{c}} & c \leq r \end{cases}$$

where c is length of the Machine Translation and r is the length of the References. P_n are different precision scores for different n-grams and w_n are the weights given to the different precision scores.

9. Results

Here are the results that we obtained assessed on different metrics:

MSR-VTT (sports)	BLUE-4	BLEU-3	METEOR
global	39.26	49.91	26.64
local	41.34	50.01	26.89

MSR-VTT (movies)	BLUE-4	BLEU-3	METEOR
global	46.76	58.91	26.94
local	47.12	59.56	28.04

MSVD (overall)	BLUE-4	BLEU-3	METEOR
global	49.11	59.12	32.99
local	50.21	60.86	33.68

Fig. 11. Results

9.1. Correct descriptions

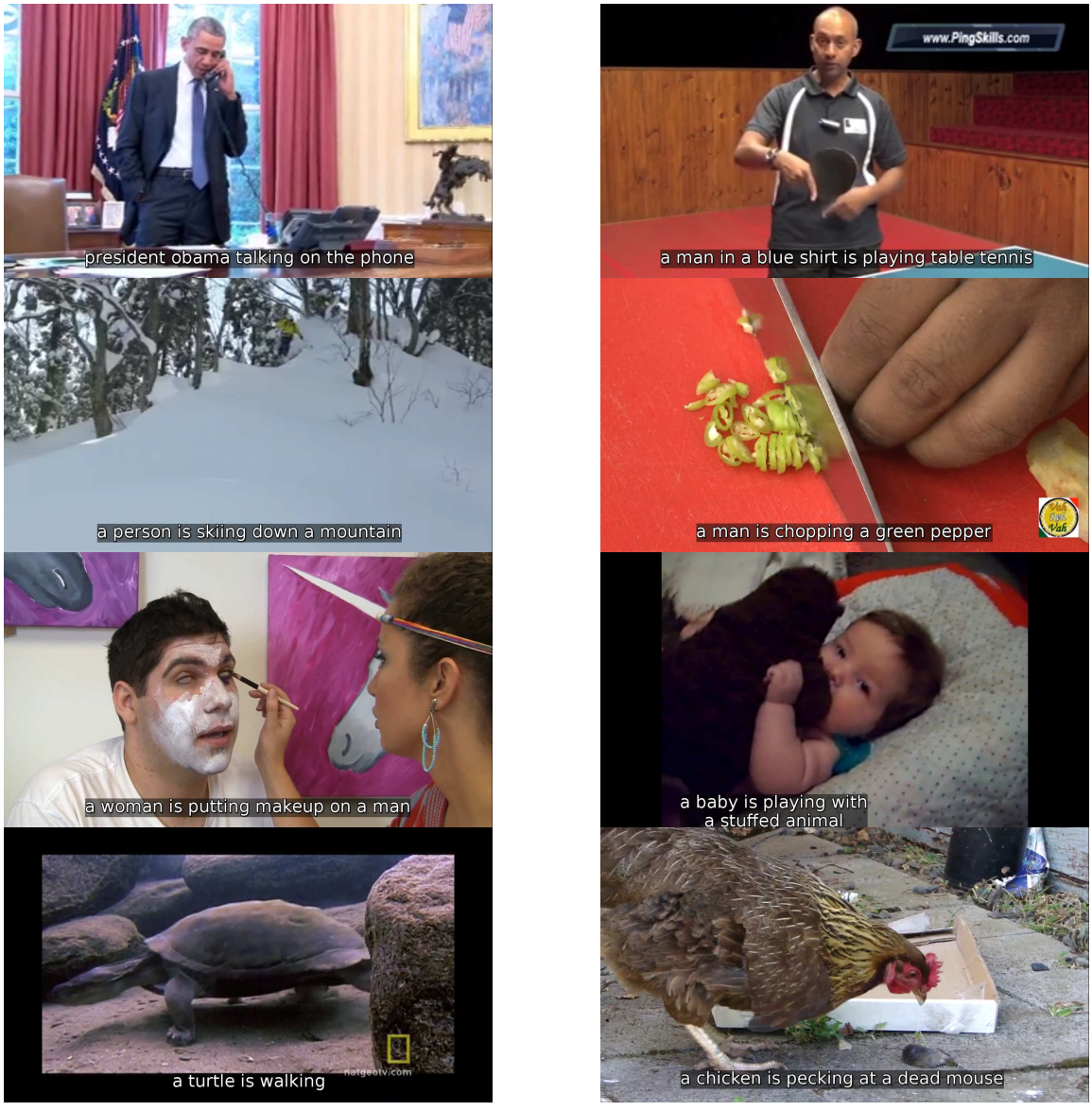

Fig. 12. Correct descriptions involving different objects and actions for several videos

9.2. Relevant but incorrect descriptions

In fig-13, we can see that the captions given by the model are not accurate descriptions of the video. In the first video a man is just posing for the camera and some people are celebrating in subsequent frames but model assumes the person is speaking something and in other a person is running on a train but the model predicts water the person is moving so it outputs that the man is running on water.

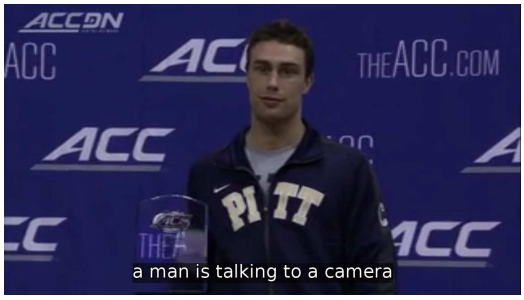

Fig. 13. Relevant but incorrect descriptions

9.3. Irrelevant descriptions

In fig-14 the captions are not relevant to the videos as in one video a person is playing and no other person is present in the video but predicts that a man is talking to another person. In second video model predicts hen as monkey and caption does not even has a meaning.

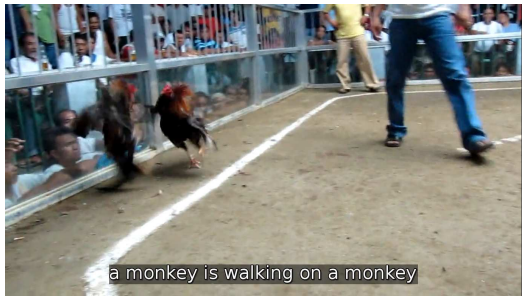

Fig. 14. Descriptions that are irrelevant to the event in the video

10. Our Contributions and Modifications to the framework

With the advantage of referring the decent amount of literature present about the topic there is not just one model or paradigm we have implemented,instead we collected the design mechanisms and derived architectural ideas from several frameworks to manifest this one single purpose.Since we have analyzed the literature we could able to inspire from different models for different purposes and functionalities.
Apart from incorporating insights from various models we had to perform one more critical task of bringing all resources together, when we bring different ideas ,resources sub-model architectures from distinct places putting them together to work as one single unified system is not that easy as they have different standardization principles ,various protocols and underlying mechanisms so we had to this side of act too changing mechanisms and standardizing them to work as a single unit.
So the complete process of collecting,deriving,modifications and standardizing the architecture will be discussed in detail in following subsections:-

10.1. Embedding different Encoder-Decoder model into the architecture

As discussed in the purpose,we are just inspired by the idea of reconstruction network not entire architecture of the model.So with the sufficient literature knowledge acquired we tried to bring our mode of architecture in the terms of convenience and efficiency.
Since reconstruction network is compatible with any encoder-decoder architecture we took a sense of liberty to exploit this opportunity to make the model more simple , convenient and efficient as

discussed.The process of modification is discussed below:-

1) In the initial model the functionality of encoder is to just pass the sequence of frames into pre-trained Convolutional networks(Inception-v4 in our case) and the send the encoded feature representations to the Decoder as context.
2) So the temporal dependencies between the frames is not captured, in order to exploit the temporal dependencies across the frames we have induced LSTM's to the encoder.
3) The input to the LSTM cells in encoder now is the encoded semantic feature representations of frames of video after passing them through pre-trained convolutional networks.
4) And the hidden states generated by LSTM's in encoder at each time step is now feeded as context to the decoder.

Essentially we are removing the encoder decoder stages of the reconstruction network and adding new variety of Encoder and decoder stages with LSTM's induced in Encoder stage.The decoder must also be altered to be be compatible with the changed encoder architecture.

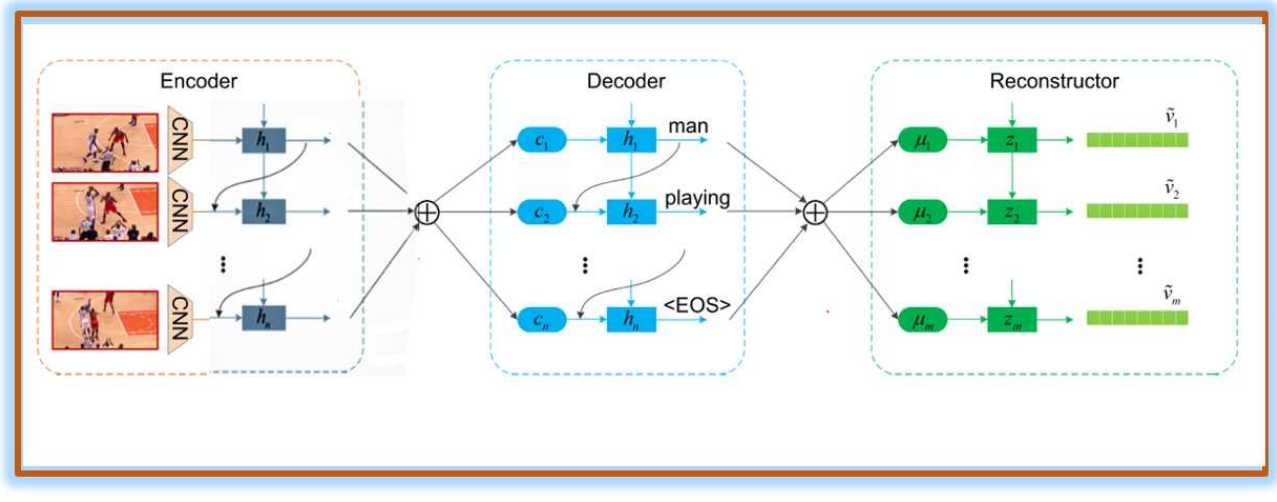

Fig. 15. Architecture of the Encoder-Decoder-Reconstructor model after inducing the new encoder decoder stages

10.2. Altering Decoder Architecture

With induction of LSTM's into encoder stage this might bring resemblance to architecture of S2VT(Sequence to Sequence – Video to Text) but the architecture we have employed is completely varies with that of S2VT.S2VT has encoder and decoder are stacked together with the shared weights but in our framework the decoder is completely decentralized with that of encoder stage.Now decoder has to to take the context not from frame representations but from the hidden states of the encoder,the attention layer has to be changed accordingly.

10.3. Modifying internal structure of Attention mechanism

As discussed in the above section to display the variation with S2VT model and conventional reconstruction network ,the S2VT model doesn't have attention mechanism between encoder and decoder but we preserve the attention layer but changing it's working structure.The attention mechanism should distribute weights for hidden state consequence generated by encoder LSTM not for semantic features of video frames .This is the consequence to cope with the major architectural change with respect to encoder.
The corresponding change is illustrated in the following figure:-

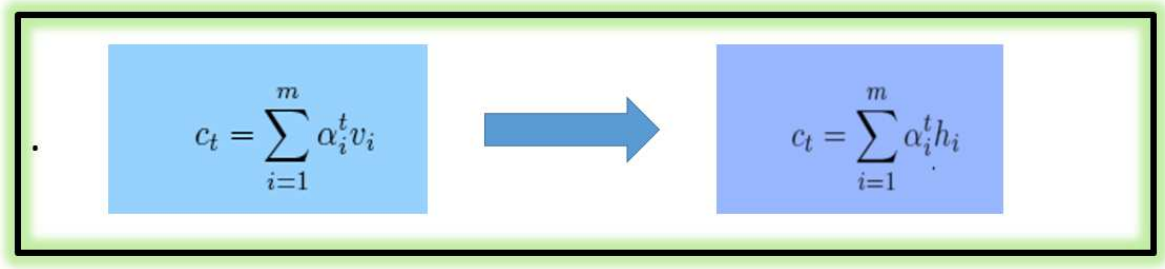

Fig. 16. Here c_t corresponds to the context vector passed to decoder based on attention mechanism.Initially the weights were distributed across v_t feature representations of the frames.Now after our modification the weights are distributed over h_t hidden states of the LSTM cells in the encoder.

10.4. Decentralizing the process of Encoding feature representations of Video frames using pre-trained networks

Usually the encoding features using pretrained convolutional networks is part of the encoding stage as these features are treated as input for the LSTM cells in temporal fashion.But due to the heavy computational constraint the process of finding features using pretrained networks is completely decentralized from the encoder stage.As the encoding process takes relatively high computational time as it has to both encode and train LSTM.
Hence we process the video frames by sampling them from the video and finding the feature representations by passing through pretrained inception-v4 network and store them in the form of hdf5(Hierarchical Data Format-5) file.Through this hdf5 file relevant features are extracted and fed to LSTM's in encoder.
This is a conscious collective pseudo-architectural decision to handle technical complexities.

10.5. Fragmenting end-end pipe line

The proposed architecture for reconstruction network is train the entire model i.e encoder-decoder-reconstructor model with the joint loss .But we made this into two stage pipe line as detailed below:-

1) **Stage-1**:- First the **Encoder-Decoder** model is trained seperately without the reconstruction network.This network is trained dedicated loss pertained to only encoder-decoder structure. So the dedicated loss used to train only encoder-decoder network is:-

$$L(\theta, \theta_{rec}) = \sum_{i=1}^{N} (-\log P(S_i|V_i;\theta))$$

2) **Stage-2**:-After training encoder-decoder network ,now **Reconstruction network** is attached to the stage-1 network to fine-tune the model to generate good hidden state representations which can embed enough semantic information about the video to decode the caption sequence.
This composite network i.e **Encoder-Decoder-Reconstructor** Network is trained using joint loss as follows:-

$$L(\theta, \theta_{rec}) = \sum_{i=1}^{N} (-\log P(S_i|V_i;\theta) + \lambda L_{rec}(V_i, Z_i;\theta_{rec}))$$

11. Generalization

To present the generalizing capability of our model we inferred the model on some external and non-related videos other than the dataset but with in the same domain. So since our primary category domain of caption generation is the area of sports we have scraped some videos of

students playing sports in **SPANDAN,sports fest of IIIT Bangalore** and generated captions on these videos.
The corresponding demonstrations are displayed in the Google drive link mentioned below.The videos are divided into two broad categories based on the model's efficiency to predict captions namely **Relevant** and **Irrelevant** categories.
So click the following google drive link for exploring them:-

Click here to view captions generated on " Generalized Videos"

12. Future Work

One possible extension that can be done for this model is to add **self-attention** across the video frames such that the relationship between the frames of the video are exploited.
This further could also be extended to incorporate multi-head attention layers with feed forward layers and masked attention at the decoder stage .And on top of this adding reconstruction network to both encoder and decoder stages.In encoder reconstruction network can be added to reproduce video frame representations from hidden state sequence of the encoder and in decoder reconstruction network to reproduce context of the encoder after attention from the hidden state sequence of the decoder.This can be compositely called as **Transformer-Reconstruction network for Video captioning**.

13. Conclusion

So here we present a best possible model we could for the task of video captioning by referring the available literature present in the community .We have gathered the different aspects from different model framework for different purposes and inspired from diverse ideas ,all these are comprehensively put together to form a composite model in order to achieve simplicity,convenience and efficiency.The primary idea that stands out is the concept of reconstruction network which has the ability to reproduce video frame representations from the hidden state sequence of decoder(which is used to generate caption sentences).So the motive behind is that when we impose this backward flow of reconstruction in the training the decoder hidden states tries to learn more and more semantic information about the video as they have the potential information to reconstruct the video itself.Altogether this becomes a complete and efficient model fo the given task of Video Captioning with dual learning or bi-directional flow of information.

14. References

Video Description: A Survey of Methods, Datasets and Evaluation Metrics

Multimodal Feature Learning for Video Captioning Sujin Lee and Incheol Kim

S. Venugopalan, M. Rohrbach, "Sequence to sequence - Video to text,"

Reconstruction Network for Video Captioning Bairui Wang, Lin Ma, Wei Zhang, Wei Liu

https://www.coursera.org/lecture/nlp-sequence-models/bleu-score-optional-kC2HD

http://cs231n.stanford.edu/reports/2017/pdfs/31.pdf

https://drive.google.com/drive/folders/1o7fSXSstE198Jauwcju5lcJWozlvjFgD?usp=drive$_{open}$