

Video Captioning Using LSTM with Attention Mechanism

Project report submitted in partial fulfillment of the requirement for
the degree of Bachelor of Technology

In

Computer Science and Engineering/Information Technology

By

Shubham Garg(161456)
Himanshu Gupta(161454)

Under the supervision of
Dr. Suman Saha

to



Department of Computer Science & Engineering and Information
Technology

**Jaypee University of Information Technology Waknaghat, Solan-
173234, Himachal Pradesh**

Candidate's Declaration

I hereby declare that the work presented in this report entitled “ **Video Captioning Using LSTM with Attention Mechanism** ” in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2018 to December 2018 under the supervision of **Dr. Suman Saha (Assistant Professor (Senior Grade))**.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.


(Signature)

Himanshu Gupta 161454

Shubham Garg 161456

The image shows two handwritten signatures. The first signature is for Himanshu Gupta, written in black ink. The second signature is for Shubham Garg, also in black ink, with the name 'garg' written in a smaller font below the main signature.

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

A handwritten signature in black ink, which appears to be 'Saha', written in a cursive style.

(Signature)

Dr. Suman Saha

Assistant Professor (Senior Grade)

Computer Science and Information Technology

Dated: 28/05/2020

Acknowledgement

We express our profound gratitude and deep regards to our project supervisor and mentor Dr. Suman Saha for his guidance and support. He took keen interest and guided and encouraged us both in our project titled “Video Captioning using Attention Based LSTM”. The guidance and mentorship shall carry us long way in our journey of life.

We would also like to thank all the staff members of Jaypee University of Information Technology, Wagnaghat for their timely support and all the information they provided.

Lastly, we would like to thank our batch mates for their motivation without which it would be impossible to complete this project.

Till the completion of our project by providing all the necessary information for developing the project. The project development helped us in research and we got to know a lot of new things in machine learning and deep learning.

Abstract

As the world is evolving at an exceptional pace, people are relying more on computers for their daily task. Artificial Intelligence is playing a major role in this evolution. People need efficiency and accuracy in their daily tasks. AI is helping people achieve this efficiency and accuracy. For instance, if you need a video to be captioned, you'll require a deep learning model trained on a huge number of videos. This model might further be used by other applications.

In this project, we train our model to extract features from frames of a video and then based on the time sequence of the frames, we train the model to evaluate what's happening in the video. For this we use MSR-VTT dataset and MSRVD dataset.

By the time we finish this undertaking, we will have a model that can be trained for any video dataset and will be able to caption the videos with a great efficiency as well as accuracy.

Table of Content

1. INTRODUCTION	1- 3
1.1 Introduction.....	1
1.2 Problem Statement	2
1.3 Objectives	3
1.4 Methodology.....	3
2. LITERATURE SURVEY... ..	4-
24	
2.1 Video captioning with recurrent networks based on frame and video level features and visual content classification	4
2.1.1 Overview.....	4
2.1.2 Model	4
2.1.2.1 Video feature extraction.....	5
2.1.2.2 Image feature extraction... ..	5
2.1.2.3 Visual content classification	6
2.1.2.4 LSTM caption generator	6
2.1.3 Conclusion	8
2.2 Video Captioning by Adversarial LSTM.....	9
2.2.1 Overview.....	9
2.2.2 Model	10
2.2.2.1 Objective Function.....	11
2.2.2.2 Generative Model	12
2.2.2.3 Discriminative Model	12
2.2.3 Results.....	13
2.2.4 Conclusion	13
2.3 Sequence to Sequence – Video to Text	14
2.3.1 Overview.....	14
2.3.2 Model	14
2.3.2.1 Sequence to sequence video to text.....	14
2.3.2.2 Video and text representation	15
2.3.3 Results.....	17

2.3.4	Conclusion	17
2.4	Video Captioning with Attention-Based LSTM and Semantic Consistency.....	18
2.4.1	Overview.....	18
2.4.2	Model.....	19
2.4.2.1	Terms and Notations.....	19
2.4.2.2	Attention-Based Long Short-Term Memory Decoder	20
2.4.2.3	Loss : Translation From Videos to Words.....	23
2.4.3	Conclusion	24
3.	SYSTEM DEVELOPMENT	25-38
3.1	Convolution Neural Network (CNN).....	25
3.1.1	Theory of CNN	25
3.1.2	Structure of CNN	26
3.2	Recurrent Neural Networks (RNN)	29
3.2.1	Theory of RNN (RNN)	29
3.2.2	Problem with a normal RNN	30
3.3	Long Short-Term Memory (LSTMs)	31
3.3.1	Theory of LSTMs.....	31
3.3.2	Structure of LSTMs.....	32
3.3.3	Working of LSTMs.....	34
3.4	Attention Mechanism.....	35
3.4.1	Theory of Attention Mechanism	35
3.4.2	Working of Attention Mechanism... ..	37
4.	PERFORMANCE ANALYSIS	39-40
4.1	Performance affected by Dataset	39
4.2	Performance improvement using LSTM	39
4.2.1	Vanishing Gradient	39

4.2.2 Exploding Gradient.....	39
4.3 Performance improvement using Attention mechanism.....	40
5. CONCLUSION.....	41
5.1 Conclusions.....	41
5.2 Future Scope	41
5.3 Applications	41
References.....	42
Appendices.....	43-45

List of Figures

1. Fig 1.1: Working of model.....	2
2. Fig 2.1: Block diagram of model	5
3. Fig 2.2: Modular Structure.....	10
4. Fig 2.3: LSTM-GAN model	11
5. Fig 2.4: Convolution process of input sentence in discriminative model	12
6. Fig 2.5: S2VT model.....	16
7. Fig 2.6: Model of aLSTM	19
8. Fig 2.7: LSTM Unit.....	20
9. Fig 2.8: Illustration of temporal attention mechanism in the LSTM decoder process	22
10. Figure 3.1: CNN Working	25
11. Figure 3.2: Inputs of CNN	26
12. Figure 3.3: Fully-connected layer.....	26
13. Figure 3.4: CNN with grouping	26
14. Figure 3.5: Multiple layered CNN	27
15. Figure 3.6: Complete CNN	27
16. Figure 3.7: 2D CNN.....	28
17. Figure 3.8: Complete 2D CNN	29
18. Figure 3.9: RNN structure.....	30
19. Figure 3.10: Information persisting in RNN	30
20. Figure 3.11: Loss in RNN	30
21. Figure 3.12: LSTM Structure.....	31
22. Figure 3.13: Repetitive structure of RNN	32
23. Figure 3.14: Linear cell state transformation	32
24. Fig 3.15: Gate.....	33
25. Fig 3.16: Input gate in LSTM	33
26. Fig 3.17: Forget Gate in LSTM	33
27. Fig 3.18: Output Gate in LSTM.....	34
28. Fig 3.19: Addition of current input and previous state	35
29. Fig 3.20: Soft Attention	36
30. Fig 3.21: Hard Attention	36
31. Fig 3.22: Working of attention mechanism	37

1. INTRODUCTION

1.1 INTRODUCTION

Depicting visual substance with text has as of late gotten amplified interest, especially portraying pictures with a single sentence. Video depiction has up to this point seen less consideration in spite of its essential applications in human-machine cooperation, video compartmentalization, and portraying motion pictures for the visually impaired.

While picture depiction handles a variable length yield arrangement of words, video portrayal conjointly should deal with an input of unknown length grouping.

Other ways for dealing with the portrayal have settled unknown length contribution by comprehensive clip portrayals, pooling over edges, or sub-inspecting on a fixed assortment of info outlines. In differentiation, during this work we will in general propose a grouping to arrangement model that is prepared start to finish and is in a situation to discover self-assertive fleeting structure inside the info succession. Our model is an arrangement to succession as in it peruses in outlines consecutively and yields words successively.

The issue of creating portrayals in open space recordings is difficult not just gratitude to the different arrangement of items, scenes, activities, and their characteristics, yet additionally in light of the fact that it is difficult to see the striking substance and depict the occasion fitly in setting. To realize what's worth depicting, the model gains from clip cuts and matches sentences that portray content imagined in the occasions.

We utilize Long Short Term Memory (LSTM) systems, a kind of repetitive/recurrent neural network system (RNN) that has made incredible progress on comparable grouping to-arrangement assignments, for example, discourse acknowledgment and machine interpretation. Because of the intrinsic consecutiveness of language and recordings, LSTMs are the best tool for creating depictions of the occasions in the recordings/clips.

Our model directly learns to map a sequence of frames to a sequence of words. A stack of LSTMs encodes the frames extracted from the video, taking as input the output of a Convolutional

Neural Network (CNN) applied to each input frame's extracted features vector. Once all frames are feed to the model, the model generates the words one by one to create a sentence. The encoding and decoding of the frame and the word representations are learned jointly.

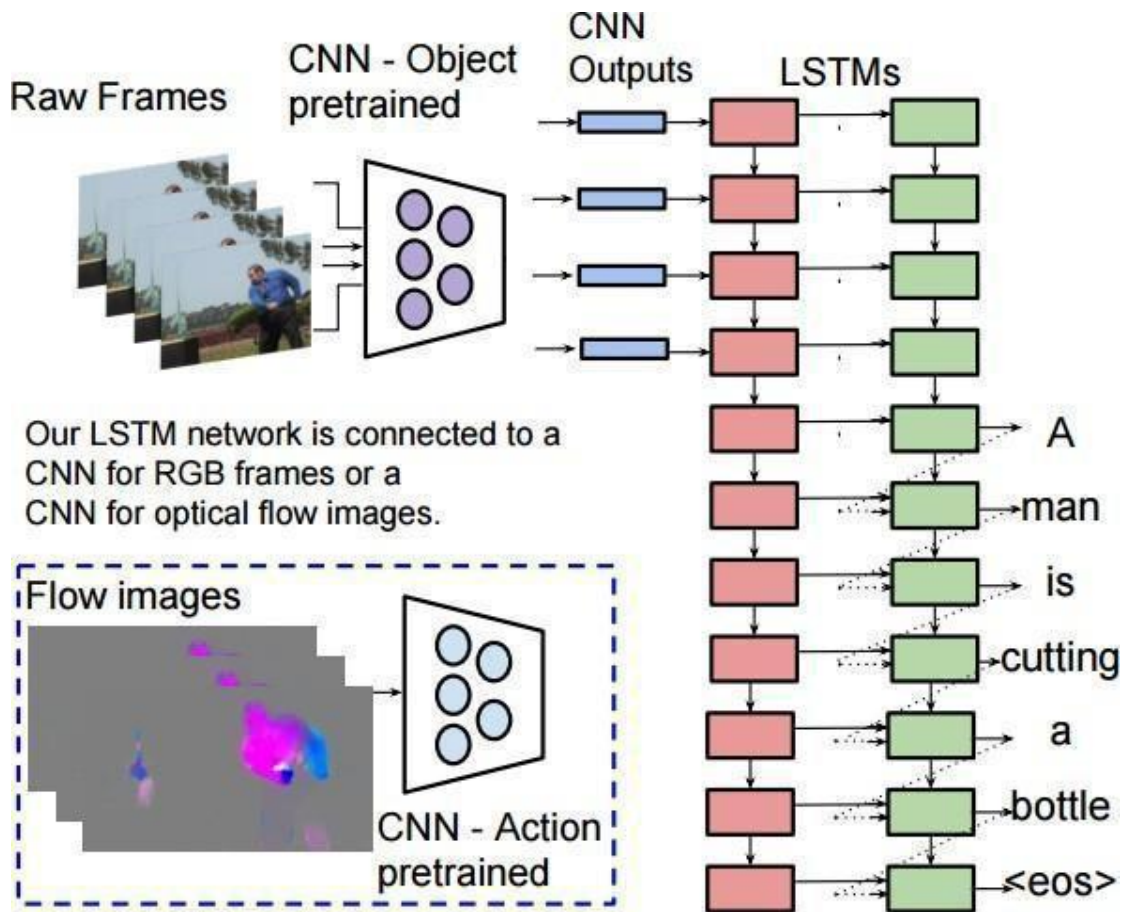


Fig 1.1: Working of the Model

1.2 PROBLEM STATEMENT

Describing an image is much easier as compared to a video, as there is more dynamic content, sequence and time dependency in a video. Our aim is to make a model that takes these issues into account. For this we are using Long Short Term Memory (LSTM) and Convolutional Neural Network (CNN) and are proposing a Sequence to Sequence model for describing a video clip.

1.3 OBJECTIVES

Trim our dataset as we don't have high computational power systems.

To generate the feature vectors using 2D convolutional neural network

To build vocabulary model

To build LSTM encoder-decoder model

To implement and add attention mechanism to the model

1.4 METHODOLOGY

We put forward a grouping to arrangement model which depict video, where the info is a succession of clip outlines (x_1, \dots, x_n) , and the yield is a grouping of words (y_1, \dots, y_m) . Both the info and yield are unpredictable, conceivably contrast in lengths.

In this case, the number of frames is higher than the number of words. In the model, we figure out the conditional probability of the output sequence (y_1, \dots, y_m) which is the sentence itself, given a input sequence (x_1, \dots, x_n) , that is -

$$p(y_1, \dots, y_m | x_1, \dots, x_n)$$

This problem is equivalent to machine translation between natural languages, where a sentence which is in one language is interpreted into a sentence of another language with the same meaning. Recent work in this field has shown the effectiveness and efficiency of LSTM Recurrent Neural Network (RNN) in solving sequence to sequence problems with precision.

2. LITERATURE SURVEY

2.1 Video captioning with recurrent networks based on frame and video level features and visual content classification [1]

2.1.1 Overview

In this paper, the author portrays the framework for producing printed portrayals of short video cuts utilizing intermittent neural networks. The research was expanding on static picture inscribing framework and stretches out this structure to recordings using fixed photo features & specific video features.

They also studied the worth of classifiers for image and videos, as a wellspring of extra data in favor of inscription age. In the midst of outcome they demonstrate so as to using the key frame based element, the thick direction video features & the substance classifier yields together gives preferred outcomes over anybody of them separately.

2.1.2 Model

They propose to utilize a neural system based structure to create subtitles for the given video. It comprises of 3 different stages. The first one is is feature extraction, where they extricate both entire video and key frames picture based features from the data sources. The features from the picture are extricated by passing these pictures to the CNN prepared on the ImageNet database.

They utilize 3 diverse CNN models for a rich assortment of features for the key frame pictures. Convolution Neural Network based features and explicitly late combination mixes are set up to give great execution during numerous computer vision as well as picture processing issues.

The third stage comprises of a LSTM that takes one list of capabilities as information and produces a relevant caption for video.

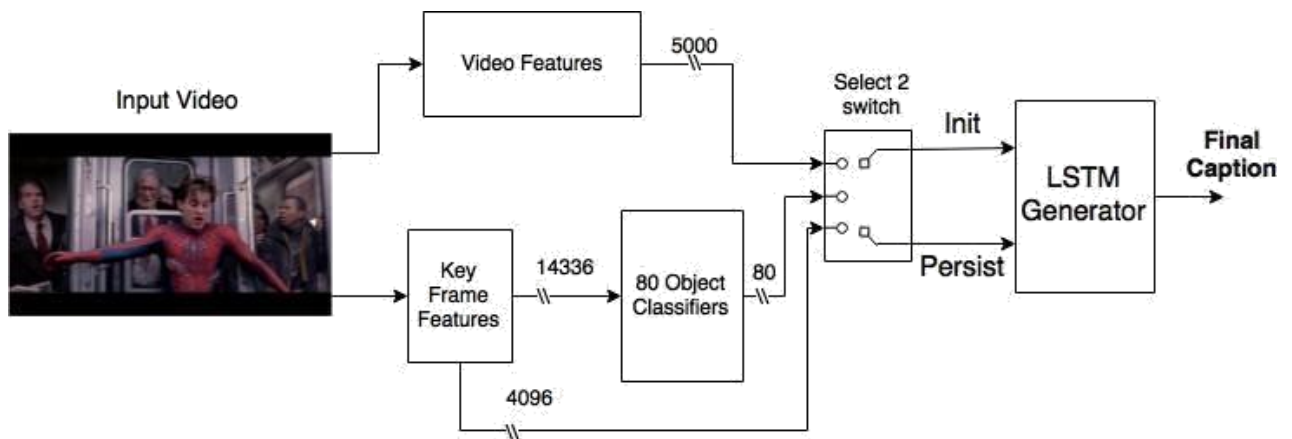


Fig 2.1: Block diagram of model

2.1.2.1 Video feature extraction

They have utilized the base length of two seconds for the video when extricating features. Recordings that had edges more noteworthy than 800 pixels were downsized to the breadth of 720 pixels. Pre set length of fifteen was utilized in the directions, giving twenty eight dimensional dislodging resultant. These were computed to a histogram with 1000 dimensions, which was then made through k-means bunching a million arbitrarily examined directions from the preparation information of the LSMDC recordings.

They likewise made thousand dimensional histograms from the ninety six dimensional Hoards, Movement Limit MBHy & MBHx descriptors plus the HOF with 108 dimensions, information in thick directions. In the wake of linking every five histograms brings about five thousand dimensional video features.

2.1.2.2 Image feature extraction

They extricate static pictures from one key frame chosen as of the middle of each video. For extraction of the components in the key frames they utilize Convolution Neural Networks pre-prepared on a database which is the ImageNet. We utilize three distinctive Convolution Neural Network designs specifically 16-layer and 19-layer VGG nets and GoogLeNet. They use VGG nets to separate initiations of that system upon 2nd completely associated fc 7 layer with 4098 dimensions in favor of the known information pictures with the perspective proportion

Contorted as 2D square. 10 districts are separated from all pictures and normal pooling of the locale shrewd highlights are utilized to produce the last highlights. For GoogLeNet we have utilized also the fifth Beginning module, having 1024 dimensions. They expand the highlights using the invert spatial pooling pyramid ace presented through 2 scale levels.

The 2nd stage dwell a 3×3 lattice with covers and even flipping, ensuing an aggregate of twenty six districts, of the size of 2. The initiations from locales are subsequently pooled utilizing normal along with most extreme pooling. The initiations from various scales at last are connected coming about to highlights with 2048 dimenions.

2.1.2.3 Classification of Visual Content

They have extricated previously depicted 5 picture based on CNN, includes likewise as of the pictures inside the COCO preparing set and prepared a classifier employing SVM for every article classifications indicated within COCO(2014).

Specifically, they used direct homogeneous bit mapped SVMs of request $d=2$ to uneven crossing point bit. Hide furthermore; we utilized 2 steps of tough downbeat mining furthermore examined 5000 downbeat models on every step. For apiece LSMDC key frames they in this way comprise fifteen output of SVM (5 highlights times starting and 2 tough downbeat trainings) which they join by means of number juggling average inside late combination organize.

For those eighty combination esteems, 1 meant for every article classification, followed by connecting to frame one enrolment class vector towards every key frame picture. The vectors we alternative partner employ contributions in the direction of the LSTM organize.

2.1.2.4 Caption generation using LSTM

The generative model for output sentence dependent upon info picture, clip/video & participation highlights, they decided the utilization of LSTM systems. Previous decision depended on two fundamental prerequisites this issue forces. 1st, the model wishes to deal with output of self-assertive length which LSTMs can do by structure.

Also, during preparing utilizing angle drop strategies the mistake signs and its inclinations need to spread far back within the time without detonating, and once more LSTMs fulfil these criteria. It contains a memory cell namely ‘m’, whose incentive on whichever time step ‘t’ is impacted by the present info ‘x’, past yield ‘y’ along with previous cell state ‘m(t-1)’. The revision of the memory esteem ‘m’ is handled utilizing the information entryway as well as the overlook door. The yield is handled utilizing the output entryway.

These entryways are actualized utilizing sigmoid nonlinearity such that they are totally separable. The info and the overlook doors the cells of LSTM can safeguard the substance of the cells memory over extensive stretches assembling it suitable to adapt large arrangements. This procedure is given below through the following equations:

$$i(t) = \sigma(W_{ixx}(t-1) + W_{iyy}(t-1)) \quad (1)$$

$$o(t) = \sigma(W_{oux}(t-1) + W_{ouy}(t-1)) \quad (2)$$

$$f(t) = \sigma(W_{fxx}(t-1) + W_{fyy}(t-1)) \quad (3)$$

$$m(t) = f(t) \cdot m(t-1) + i(t) \cdot \tanh(W_{mxx}(t) + W_{myy}(t-1)) \quad (4)$$

$$y(t) = o(t) \cdot m(t) \quad (5)$$

We include a layer of softmax next to the yield of the LSTM to produce likelihood dissemination upon the jargon. At each timestep, LSTM is prepared in allocating the most noteworthy likelihood towards the word it thought off, ought to show up next given the present sources of info and the hidden state:

$$p(w_t | w_{t-1}, \dots, w_0, V) = \text{softmax}(y(t)) \quad (6)$$

In our most straightforward engineering, visual highlights are contribution to the LSTM just on the 0th time step because the information ‘x(0)’. They allude in the direction of this element contribution towards their component since it introductory sizes the shrouded LSTM’s condition.

The beginning image pursued by means of the word embedding intended for every word inside the subtitles for reference are sustained throughout information ‘x(t)’.

In the examinations in the company of the dataset which is COCO, they had thought that giving the LSTM a chance for approaching these highlights all through the age procedure will be useful which require addition of another information to the LSTM cell which allude to seeing that the determined

highlights. The information assumes a comparative job as $x(t)$ in conditions (1) – (4) aside from with an alternate arrangement of loads.

Note that we can include various highlights and tireless lines subsequently enabling the model to gain at the same time from two diverse sources. The preparing methodology for the LSTM generator, we attempt to amplify the log likelihood as-marked to the preparation tests by the model. We can decrease the negative log likelihood given as:

$$L(w_{1:L}|V) = -N \sum_{i=1}^L \log(p(w_i|w_{i-1}, V)) \quad (7)$$

2.1.3 Conclusion

Utilizing the classifier yield highlights to instate the LSTM and video includes after the introduction brings about the best execution.

Bar size one in the sentence age process is superior to anything bigger pillar sizes.

2.2 Video Captioning by Adversarial LSTM [2]

2.2.1 Overview

In this paper, they propose a novel way to deal with video subtitling dependent on ill-disposed learning and long transient memory (LSTM). They target to make up for the deficiencies of LSTM-based video inscribing strategies that for the most part demonstrate potential to successfully deal with fleeting nature of video information when producing inscriptions yet additionally ordinarily experience the ill effects of exponential mistake aggregation.

Specifically, we receive a standard generative ill-disposed system (GAN) engineering, described by an interchange of two contending forms: a "generator" that creates printed sentences given the visual substance of a video and a "discriminator" that controls the exactness of the produced sentences.

The discriminator goes about as a "foe" around the generator, and with its controlling instrument, it causes the generator to turn out to be increasingly precise. For the generator module, we take a current video subtitling idea utilizing LSTM arrange. For the discriminator, we propose a novel acknowledgment specifically tuned for the video inscribing issue and taking both the sentences and video includes as information.

This prompts our proposed LSTM–GAN framework engineering, for which we show tentatively to significantly outflank the current strategies on standard open datasets.

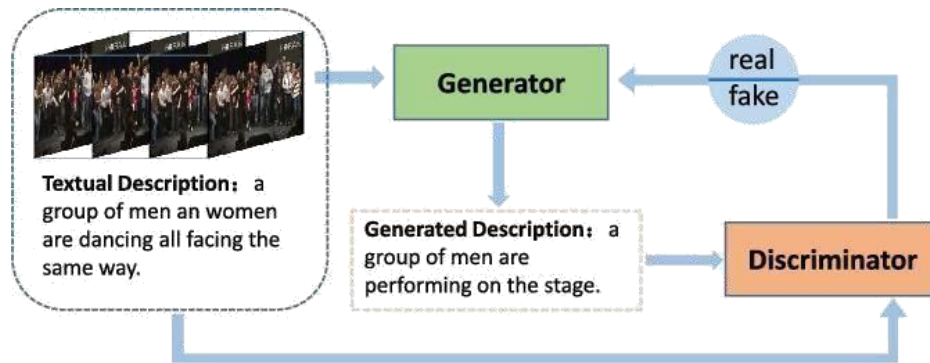


Fig 2.2: Modular Structure

2.2.2 Model

They utilized the idea of ill-disposed frameworks to gadget their model for producing video subtitles. The general structure comprises of a discriminative model D and generative model G. G is an arrangement to succession model which discloses to us how the grouping of words is produced.

D is a parallel classifier that accept contributions as succession of sentences and gives a name as a yield which lets us know whether the sentence is sensible, linguistically and normal right.

Various variations of the model were utilized to contrast and strategies effectively present.

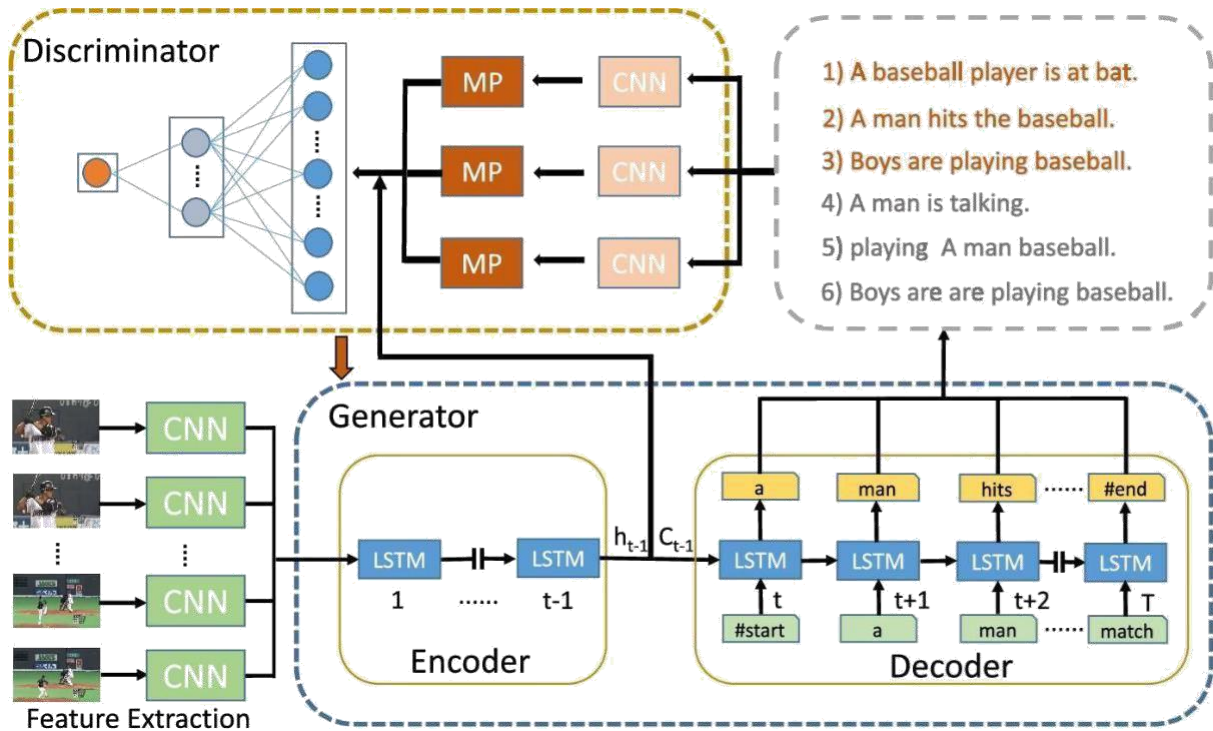


Fig 2.3: LSTM-GAN model

2.2.2.1 Objective Function

For faster convergence to the objective they pre-trained the generative G and discriminative D models. Calculating the conditional probability is the goal of G i.e. $p(S|V)$ where $V = (v_1, v_2, \dots, v_t)$ is an input sequence of frames and $S = (w_1, w_2, \dots, w_{t_1})$ descriptive texture as the corresponding output. t and t_1 represent the lengths.

$$p(S|V) = p(w_1, w_2, \dots, w_{t_1} | v_1, v_2, \dots, v_t)_{t_1}$$

$$= \prod p(w_i | V, w_1, \dots, w_{i-1})$$

Formalization of objective function D can be shown as cross entropy loss i.e. -

$$LD(Y, D(S)) = - \frac{1}{m} \sum [(Y_i) \log(D(S_i)) + (1 - Y_i) (\log(1 - D(S_i)))]$$

Where m is the count of examples in the batch, Y_i represent the real label and $D(S_i)$ represents predicted value of discriminator.

2.2.2.2 Generative Model

They utilized encoder-decoder LSTM as the generative model. Encoder is utilized to encode the highlights to fixed length vectors and the sentences are gotten in the wake of interpreting these vectors with the assistance of the decoder.

They utilized VGG16 design to outline arrangement of casings $V = (v_1, v_2, \dots, v_t)$ to an element grid. $W_v \in \mathbb{R}^{D_d \times D_t} = (w_{D1}, \dots, w_{Dt})$. D_d indicate the elements of a component vector and D_t signifies the quantity of edges.

2.2.2.3 Discriminative Model

In this model the primary target is to augment the likelihood of appointing the mark to both the preparation sentences and the once produced by G. They pick CNN as their discriminator.

The discriminator contains a convolutional layer and max pooling tasks which are utilized to catch the most extreme valuable highlights.

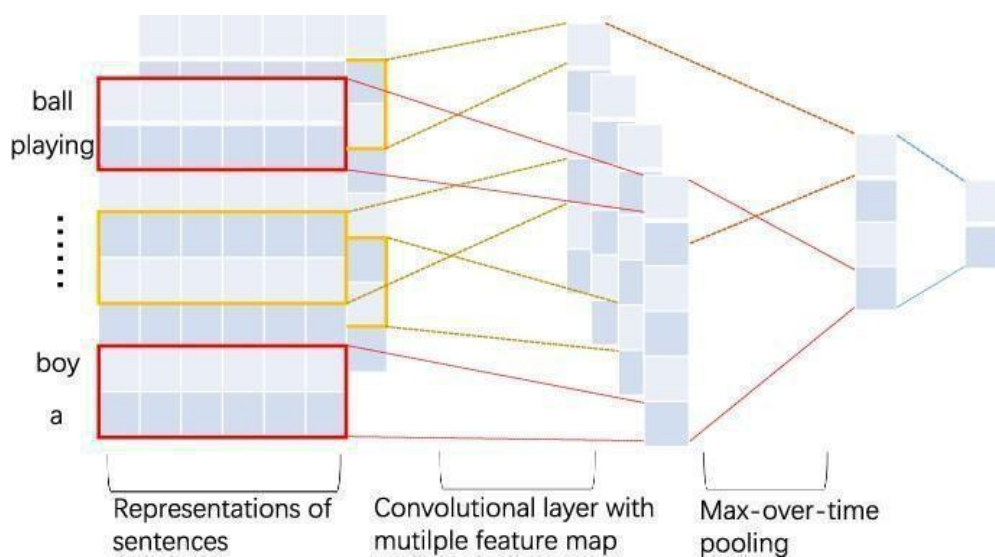


Fig 2.4: Convolution process of input sentence in discriminative model

2.2.3 Results

The outcomes of assessment measurements reliably show that our proposed LSTM-GAN accomplishes preferable execution over all the current systems.

2.2.4 Conclusion

In this paper, we exhibited a first endeavour to present the idea of antagonistic learning in taking care of the video catching issue. We accept that this idea can possibly significantly improve the nature of the subtitles, which is because of its capacity to more readily control the catch age process. This control is for this situation done by the discriminator module, going about as a foe to the subtitle age module.

Notwithstanding making the basic ill-disposed learning structure dependent on the GAN worldview reasonable for managing discrete generator yields, with our novel acknowledgment of the discriminator, we further improved the control component. This was accomplished by making the contribution to the discriminator multimodal. Along these lines, the sentences leaving the generator were approved for syntactic accuracy, yet in addition for their significance to the video content.

The capability of our LSTM-GAN system to improve the quality and assorted variety of subtitles was likewise exhibited tentatively, through an expand test study including various pattern draws near, four well known datasets, and two generally utilized assessment measurements. We accept that the presentation of LSTM-GAN could additionally be improved by depending on Fortification Learning. Support Learning has demonstrated successful for assignments like video subtitling, as for example discourse age.

2.3 Sequence to Sequence – Video to Text [3]

2.3.1 Overview

They proposed a novel start to finish succession to-grouping model to produce inscriptions for recordings. For this we misuse intermittent neural systems, specifically LSTMs, which have shown best in class execution in picture subtitle age. The Long Short-Term Memory model that is being used taught employing various clip and sentence tuples and it helps in relating edges of clip frames towards the corresponding set of words. This LSTM model can normally become familiar with the fleeting structure of the arrangement of edges just as the grouping model of the produced sentences, for example a language model.

2.3.2 Model

An arrangement to succession model for description of video is proposed. It groups video outlines, and then yields groups of words. Normally, the info and yield have a varying, possibly unique, length. Here, the number of edges is a lot larger as compared to words. This model estimates the conditional probability of an output sequence.

2.3.2.1 Sequence to sequence video to text

Initially the info grouping to a vector of fixed length utilizing one Long Short-Term Memory network is encoded and afterward utilize one more LSTM network to outline vector into a succession of yields, this model depends on a solitary LSTM network for the encoding as well as disentangling stage. This permits sharing of parameters between the encoding and unraveling stage. This model uses a pile of 2 LSTM networks with a thousand concealed units of each network. When any 2 LSTM networks are stacked together, the shrouded portrayal (h_t) from the first LSTM layer is given as the info (x_t) to the second LSTM. The top layer of LSTM in our design gets utilized in demonstrating the visual casing succession, and the following layer gets utilized for showing the yield word grouping. Preparing and Surmising in the first few time steps, the top layer of LSTM network gets an arrangement of casings and gets them encoded, and the second layer of LSTM network gets the concealed portrayal (h_t) and connects it to the invalid

to the invalid cushioned information words that it encodes at that point. There is no misfortune in this phase when encoding is being done. After every one of the casings present in the video cut gets depleted, the second layer of LSTM is bolstered the start of sentence (<BOS>) tag, which prompts it to begin interpreting its current shrouded portrayal to an arrangement of words. While preparing the unraveling stage, this model augments for the log-probability of the anticipated yield sentence given the shrouded portrayal of the video edge succession, and the past words it has seen.

Model with parameters θ and output sequence $Y = (y_1, \dots, y_m)$, this is formulated as:

$$\theta^* = \operatorname{argmax}_{\theta} \sum \log p(y_t | h_{t-1}, y_{t-1}; \theta)$$

This log-probability is enhanced over the whole preparing dataset utilizing stochastic angle plummet.

The misfortune is registered just when the LSTM is figuring out how to interpret. Since this misfortune is engendered back in time, the LSTM figures out how to create a suitable concealed state portrayal (h_n) of the info grouping. The yield (z_t) of the second LSTM layer is utilized to acquire the transmitted word (y). We apply a softmax function to get the probability distribution over the words y' in the vocabulary V .

2.3.2.2 Video and text representation

Like past LSTM-based picture inscribing endeavors and video-to-content methodologies, a convolutional neural system is applied to include pictures, and provide the yield from the top layer to the Long Short-Term Memory unit. Convolutional Neural Networks that are pretrained on the 1.2M picture ILSVRC-2012 article classification subset of the ImageNet dataset are used. Every info video outline is scaled to a 256x256 grid, and is edited to an irregular 227x227 district.

Then, it gets trained by the Convolutional Neural Network. The first last completely associated classification layer is evacuated and gain proficiency with another direct implantation of the highlights to a five hundred dimensional space. The low measurement highlights structure the information to the

first Long Short-Term Memory layer. The loads of the implanting are adapted together with the LSTM layers during preparation.

In this consolidated model, a combination system for coordinating the flow and RGB highlights is utilized. Every time at the venture of the deciphering stage, this model suggests lot of up-and-comer words. Then these speculations with the weighted whole of the scores by the flow and RGB systems are rescored, and there is a need to recompute every new word's $[p(y_t = y')]$ score:

$$\alpha \cdot \text{prgb}(y_t = y') + (1 - \alpha) \cdot \text{pflow}(y_t = y')$$

the hyper-parameter α is trained on validation dataset.

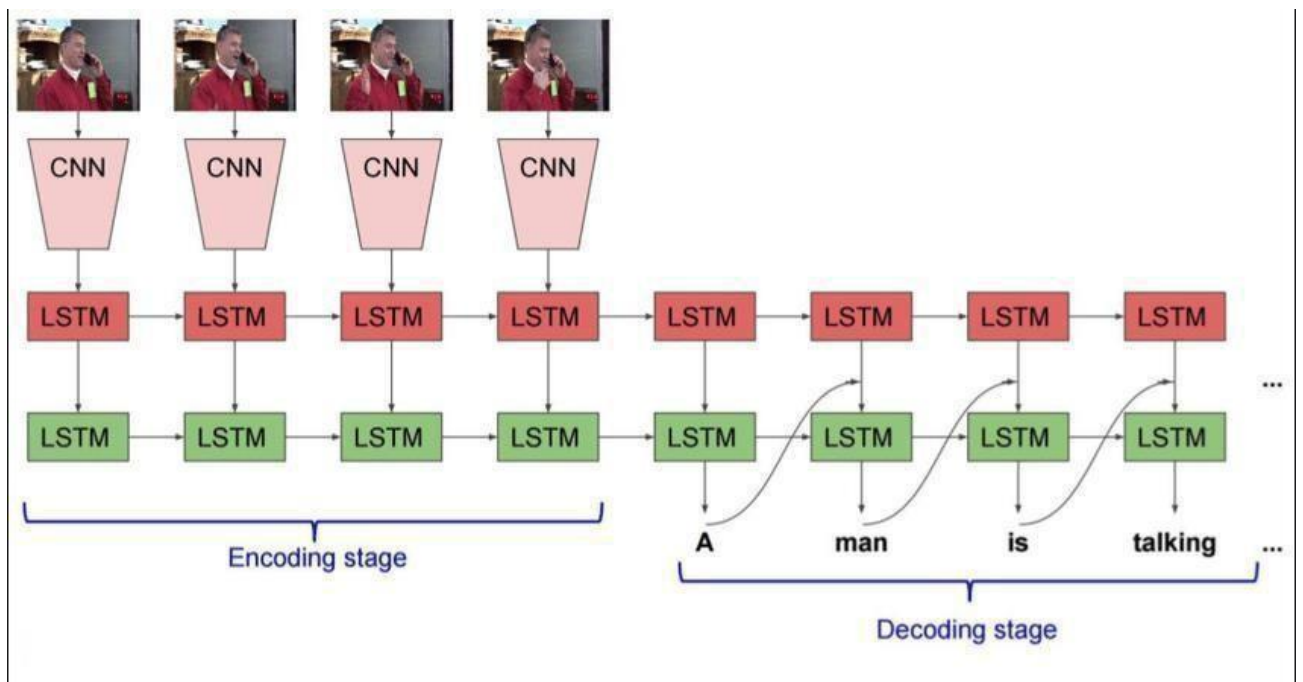


Fig 2.5: S2VT model

2.3.3 Results

This essential S2VT AlexNet model on RGB video outlines accomplishes roughly twenty-eight percent METEOR and revamps over the mean-pooled show just as the VGG mean pooled model; recommending that S2VT is an all the more dominant methodology. At the point when the model is prepared with arbitrarily requested edges, the score is extensively lower, unmistakably exhibiting that this S2VT AlexNet model gets an edge from abusing fleeting structure. This S2VT model accomplishes just 24.3% of METEOR and yet is able to improve the exhibition of the VGG model from METEOR of 29.2% to a METOEOR 29.8%, on combination. One of the purposes behind low execution of the flow model can be the optical flow includes in any event, for a similar action can differ significantly with setting for example 'panda eating' versus 'individual eating'.

2.3.4 Conclusion

Rather than related work, we develop depictions utilizing a grouping to arrangement model, where casings are first perused successively and afterward words are produced consecutively. This enables us to deal with variable-length information and yield while all the while displaying fleeting structure. This model accomplishes best in class execution on the Microsoft Research Video Description Corpus, and beats associated work on 2 enormous and testing film depiction datasets. In spite of the reasonable effortlessness, this model quite importantly gets advantage from extra information and can learn quite complicated transient structure present inside the info, yield groupings to test datasets of movie description.

2.4 Video Captioning with Attention-Based LSTM and Semantic Consistency [4]

2.4.1 Overview

By accepting a video as a succession of highlights, a Long Short-Term Memory model is prepared on pairs of videos and sentences and figures out how to relate a video with the corresponding sentence. Be that as it may, most existing techniques pack a whole clip otherwise edge into a fixed portrayal, with no taking into account of the consideration system that takes into consideration choosing striking highlights. Moreover, existing methodologies generally model the interpreting mistake, yet disregard the connections between the output sequence semantics as well as the visual substance.

For handling the stated issues, they suggest an original start towards finish structure namely aLSTMs, a consideration included LSTM model plus the semantic consistency, so that recordings can be moved to normal output sequence. This system incorporates consideration component alongside LSTM towards catching notable video structure, as well as in investigating relationship among multimodal portrayals (that is, visual & word content) that creates output with very high semantic substance.

Foremost they recommend a consideration component that uses the 2D convolutional system with dynamic weights. At that point, visual highlights are feed to LSTM used as a decoder at 't' time as well as the word-implanting highlight on 't-1' time towards generating valuable words. They finally employ multimodal inserting towards outlining the sentence & visual highlights in a common space towards ensuring semantic consistency of the output sequence portrayal plus clip visual substance.

Upon analyses on standard datasets exhibit that their strategy utilizing lone element is accomplishing aggressive in other words, far enhanced outcomes compared to cutting edge baselines for clip inscribing in METEOR as well as BLEU.

2.4.2 Model

Their undertaking was to produce sentences towards recordings. They first defined the documentations as well as the terms. Then, they present their approach i.e. aLSTM. A target work was worked through coordinating 2 misfortune capacities that at the same time think about video interpretation as well as the semantics. One misfortune work means towards ensuring interpretation from recordings to words, as well as one more misfortune work attempts towards connecting semantic hole by semantic see through relationships. The point by point data about arrangement is given too.

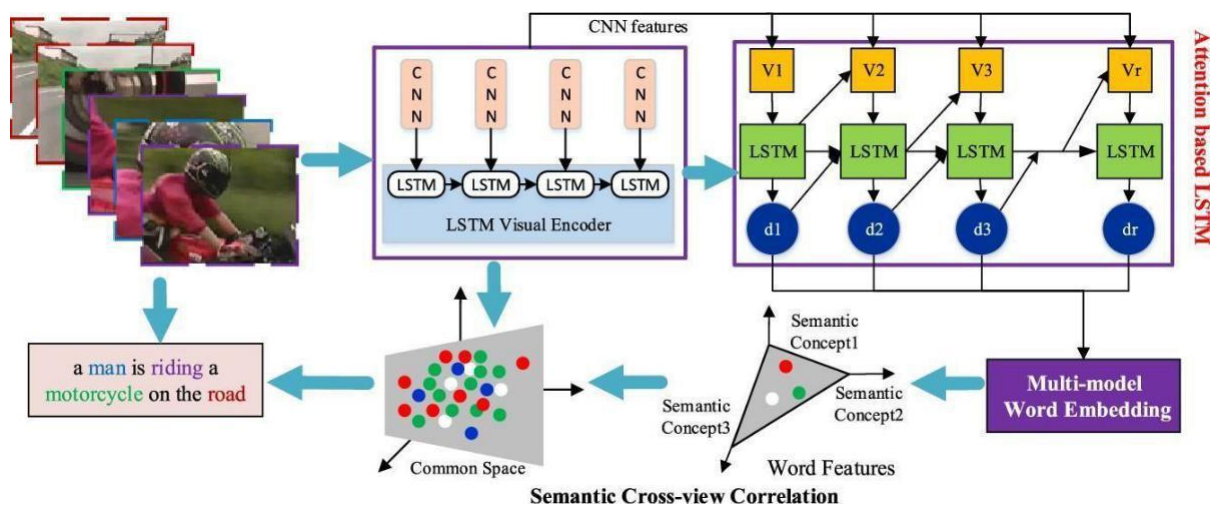


Fig 2.6: Model of aLSTM

2.4.2.1 Notations and the terms used

They assumed a clip V that is depicted as one printed output $D = \{d_1, \dots, d_{Nd}\}$ consisting of N_d words. Let $X = \{x_1, \dots, x_{Nx}\} \in \mathbb{R}^{M \times Nx}$ and $D = d_1, \dots, d_{Nd} \in \mathbb{R}^{L \times Nd}$ indicates the visual and the literary highlights, where $d_i = E d_i$ is the word portrayal of a solitary word d_i and N_x represents the count of highlight vectors.

Give L as well as M a chance to mean the component of visual element and printed include. X is extricated utilizing profound neural systems, which will be depicted in the examination.

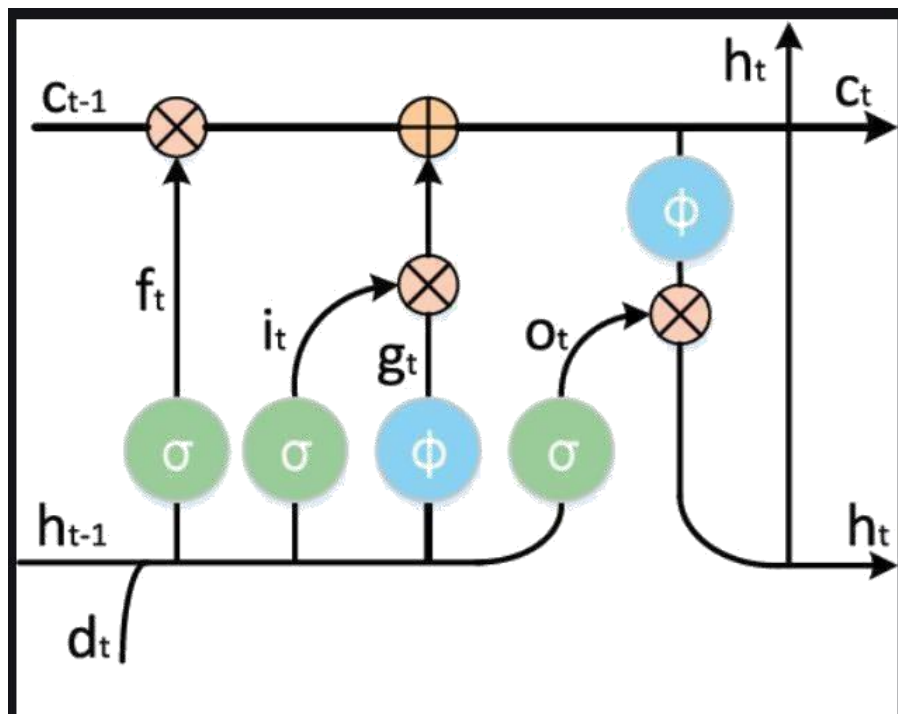


Fig 2.7: LSTM Unit

2.4.2.2 Attention-Based Long Short-Term Memory Decoder

Until now, demonstrating arrangement information with intermittent neural system has been demonstrated effective during the time spent machine interpretation, discourse acknowledgment, picture/video subtitling and so forth. Be that as it may be difficult towards preparing one regular RNN because of the one issue i.e. vanishing angle.

A refreshed form of a regular RNN i.e. LSTM, tackled the above stated issue through learning designs along more fleeting conditions. While recordings furthermore, normal outputs are equally successive information, LSTM is used like a fundamental part of their aLSTMs.

Principle thought at Consideration revolved Long Momentary Memory because it incorporate consideration system inside LSTM. An essential entity of LSTM comprises of a solitary memory cell, an information enactment capacity, and three doors (input it, overlook f_t and yield o_t). it enables approaching sign to adjust the condition of cell of memory otherwise square it. f_t handles which part to recollect & which part should be overlooked according to the cell plus some way or another

can maintain a strategic distance from the slope from disappearing or detonating when back proliferating through time. At last, it permits the condition of the cell's memory towards having an impact on different neurons else counteract it.

Fundamentally, the unit's memory as well as entryways inside LSTM square is seen as pursues:

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \phi \end{pmatrix} Z_{L+r,r} \begin{pmatrix} \mathbf{E}d_{t-1} \\ \mathbf{h}_{t-1} \end{pmatrix}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \phi(\mathbf{c}_t)$$

where E signifies an installing grid, speaks to the strategic sigmoid non-straight initiation work mapping them to (0,1) that can be seen as handles that LSTM figures out how it should specifically overlook the memory of his own or acknowledge present information, indicates a hyperbolic digression work of tanh, which astute item along the door esteem, $Z_{L+r,r}$ | signifies the LSTM's parameters. Give r & L a chance to mean installing what's more, LSTM dimensionality individually.

Contrasted and pictures, recordings contain increasingly complex fleeting data which ought to be adjusted to language information. In this way we broaden the consideration system acquainted by with help video inscribing. The new type of LSTM is defined as:

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \phi \end{pmatrix} Z_{L+r+M,r} \begin{pmatrix} \mathbf{E}d_{t-1} \\ \mathbf{h}_{t-1} \\ \mathbf{v}_t \end{pmatrix}$$

$$\mathbf{v}_t = \sum_{i=1}^{N_x} \beta_i^t \mathbf{x}_i, \quad s_i^t = W_s \phi (W_h \mathbf{h}_{t-1} + W_x \mathbf{x}_i + b_s)$$

$$\beta_i^t = \frac{\exp(s_i^t)}{\sum_{k=1}^{N_x} \exp(s_k^t)}, \text{ s.t., } \sum_{i=1}^N \beta_i^t = 1$$

Where \mathbf{v}_t speaks to setting vector that is one unique portrayal for applicable portrayal for the clip contribution at 't' time. M , which is \mathbf{v}_t 's element. Moreover, t_i is the consideration loads on 't' time depicting pertinence towards the i^{th} highlight inside information clip.

Preranged the past shrouded condition \mathbf{h}_{t-1} for the decoder plus the i^{th} clip highlight; the restoration of the normalized pertinence gain s_{t_i} . As moving forward the significance gains of every highlight $X=x_1, \dots, x_{N_x}$ are figured, the LSTM can acquire t_i at each time step t . The W_s , W_h , W_x and b_s are the parameters to be evaluated.

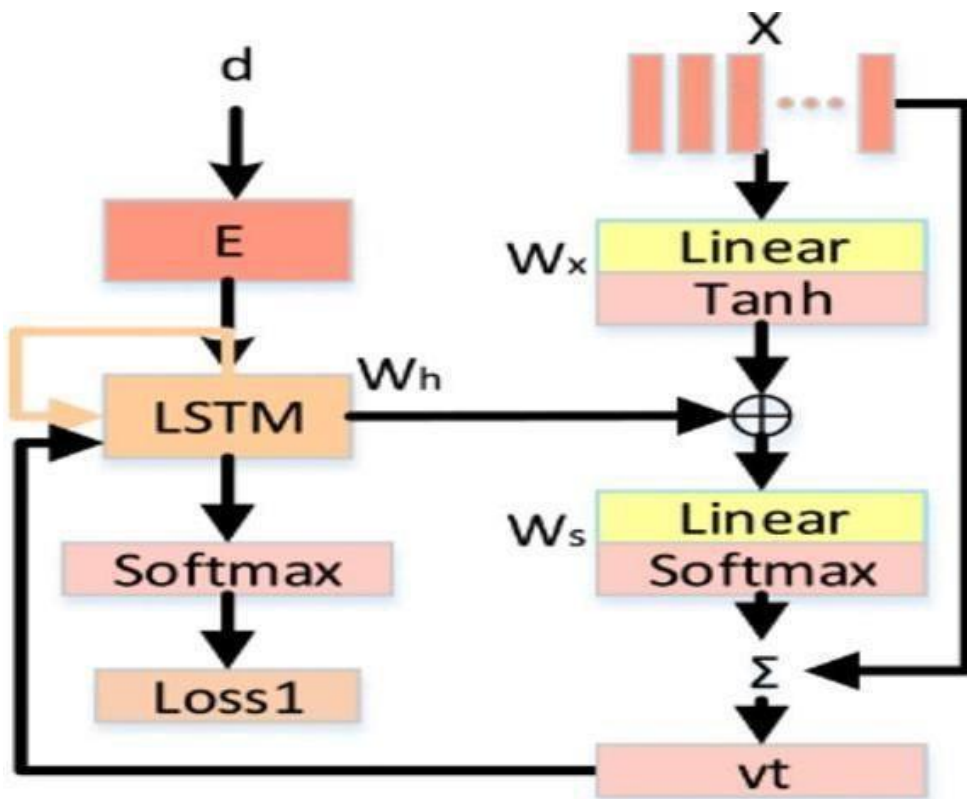


Fig 2.8: Explanation of attention mechanism inside LSTM decoding process

Moreover, to catch rich fleeting data, they presented a LSTM visual encoder with 1 layer, namely LSTM Visual Encoder. LSTMs as of late have made an incredible achievement during the time spent activity acknowledgment.

Motivated by the above stated, in their structure they recommend towards incorporation of refreshed GoogleNet along the LSTM visual encoder with 1 layer to encode clip fleeting data. Purposely, latest yield of the Visual Encoder that is the LSTM is utilized towards introduction the 1st LSTM unit of their consideration LSTM system that encourages clip subtitling.

2.4.2.3 Loss: transformation from Clips towards Words

Inside LSTM unraveling stage, the LSTM registers setting vector v_t provided a input succession $X=x_1, x_2, \dots, x_N$ and the shrouded state h_t . Motivated by the guideline of interpreting pictures, we treat the actuation esteem recorded by a preparation word d_t inside softmax layer of their output sequence generator like the probability of creating that word:

$$P(d_t | v_t; d_1, d_2, \dots, d_{t-1}; E)$$

The expense of creating that preparation word is given by the minus one times logarithm of the probability. In addition they define expense of producing the words as $Loss_1 = -\sum_{t=1}^N \log(P(d_t | v_t; d_1, d_2, \dots, d_{t-1}; E))$ where N_d means the all out total count of words inside a output sentence plus d_i indicates the i^{th} word in sentence D .

Via limiting $Loss_1$, the relevant association of the words inside the sentence be able to be ensured, creating the sentence rational as well as smooth.

2.4.3 Our understanding

They have suggested one system called aLSTMs that is executed via all the while limiting the pertinence misfortune as well as cross view loss of semantics.

On 2 well known clip depiction dataset collections, aftereffects of their examinations show how accomplished their methodology is, that accomplishes practically identical or even unrivaled execution contrasted and the present best in class models. Later on, we will change our model to chip away at space specific datasets, e.g., films.

3. SYSTEM DEVELOPMENT

Videos are nothing but a set of time dependent image frames. In case of video captioning, this time dependency is extremely important because frames only in right sequence will let us know what is happening in the video. Hence, we need to store previous inputs in order to generate an output. So, firstly we need to extract features from the frames. Algorithm which we use to do this is CNN. Secondly, we need to store these features of different frames in a timely order. Hence, we require an algorithm that has a memory. Therefore, we use LSTMs. We also make use of attention mechanism to make the system more efficient.

3.1 Convolution Neural Network (CNN)

To extract features from frames, we need an algorithm that analyses the images and identifies the relevant information from it.

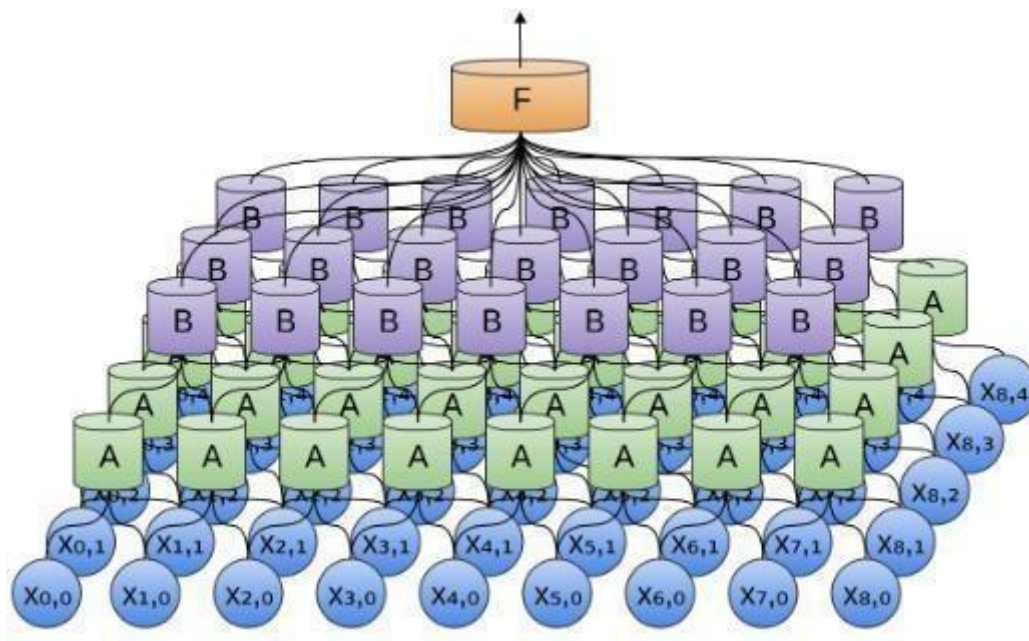


Figure 3.1: CNN Working

3.1.1 Theory of CNN

CNN or Convolution Neural Network is a deep neural network used to analyse images and extract features from them. In CNN, each neuron in one layer is connected to every neuron in

the next layer. It is a type of neural network that makes use of copies of the same neuron. It has one input layer, one output layer and intermediate hidden layers. Using relevant filters, it captures temporal and spatial dependencies.

3.1.2 Structure of CNN

Firstly, we take n inputs that are evenly spaced samples.



Figure 3.2: Inputs of CNN

The easiest way to classify them is to connect them with a fully-connected layer. Each input gets connected to each neuron.

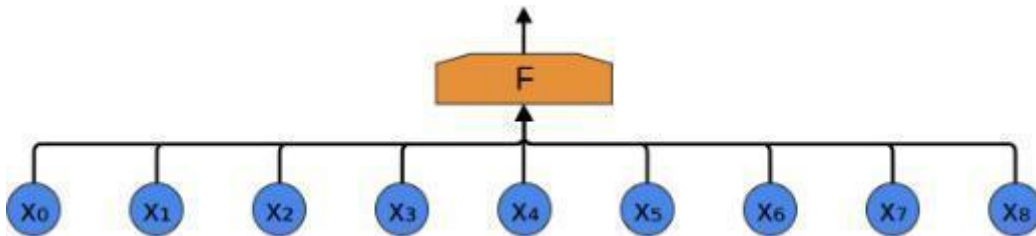


Figure 3.3: Fully-connected layer

A better approach is to notice a pattern in the properties. So, instead of directly feeding the inputs to the layer F , we first group the data. This group A computes certain features. But, it only looks at two points. In reality, we need to look at multiple points.

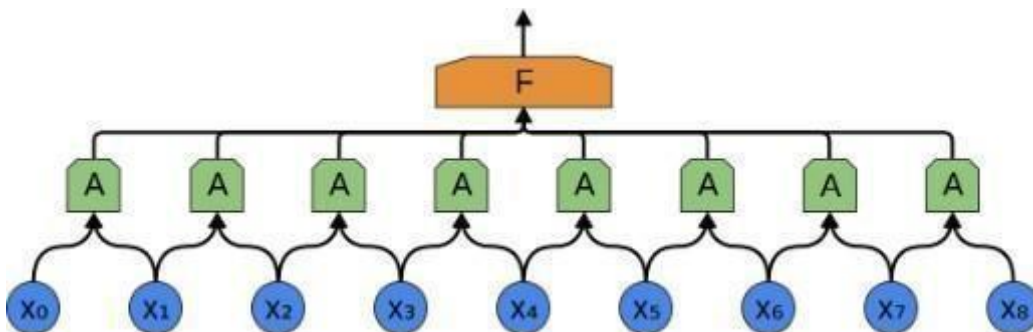


Figure 3.4: CNN with grouping

But, CNNs have a property that we can feed one layer into another making a bigger network. The complete network can therefore detect more abstract features. Adding more layers helps in looking at multiple points.

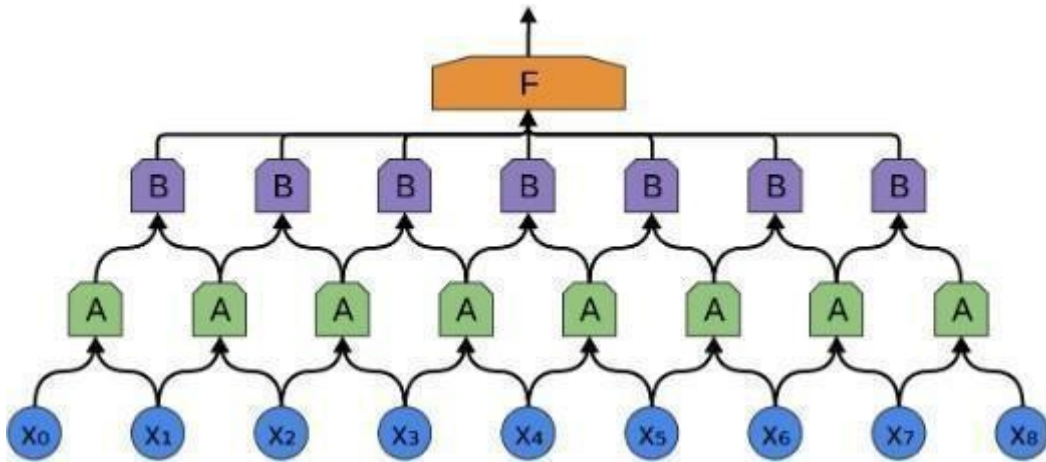


Figure 3.5: Multiple layered CNN

These convolution layers are interlaced with different kind of layers known as pooling layers. These layers pick the maximum of features from a small patch of data from the previous layers. The output of these layers predict whether a feature was present in a given region or not. After this the further layers of convolution works on a larger patch of data.

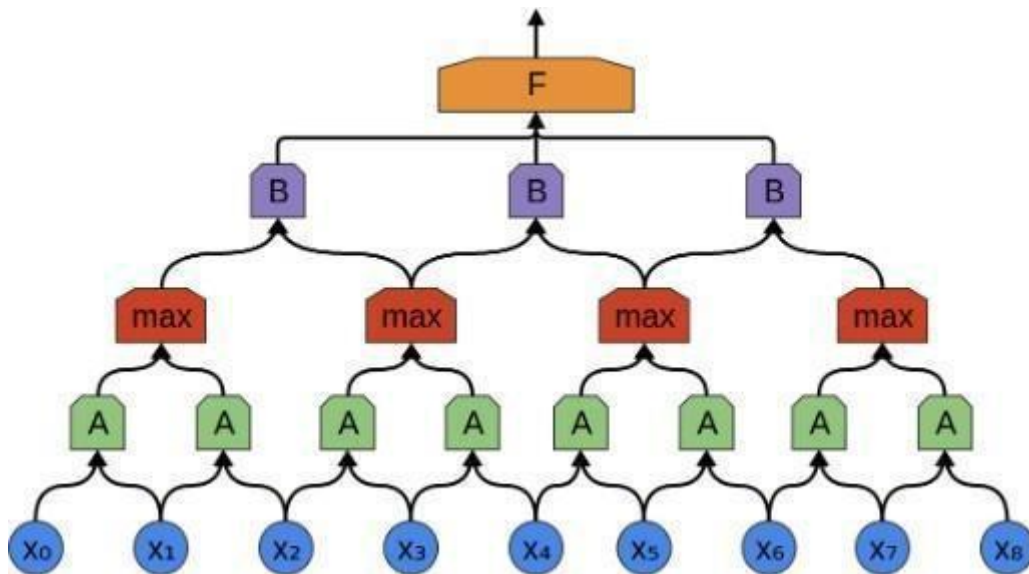


Figure 3.6: Complete CNN

The images require a two dimensional convolution layer because we have to look at patches rather than segments. The layer A computes feature for every patch, like we need to find an edge in an image or some kind of texture.

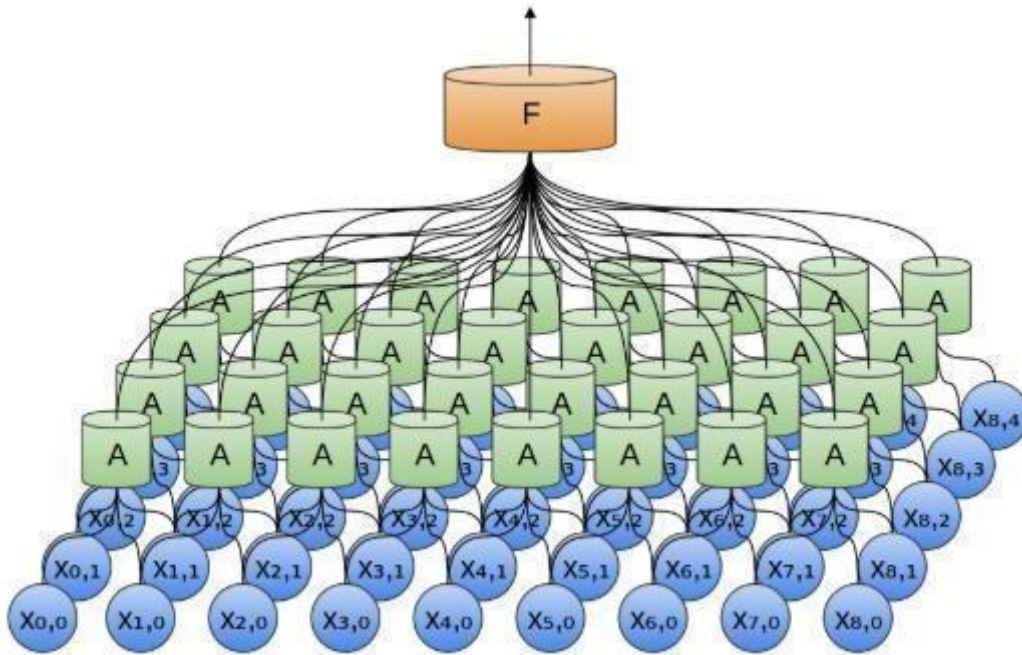


Figure 3.7: 2D CNN

Now, feeding such convolution layer to another and using pooling generate a structure like *Figure 3.8*.

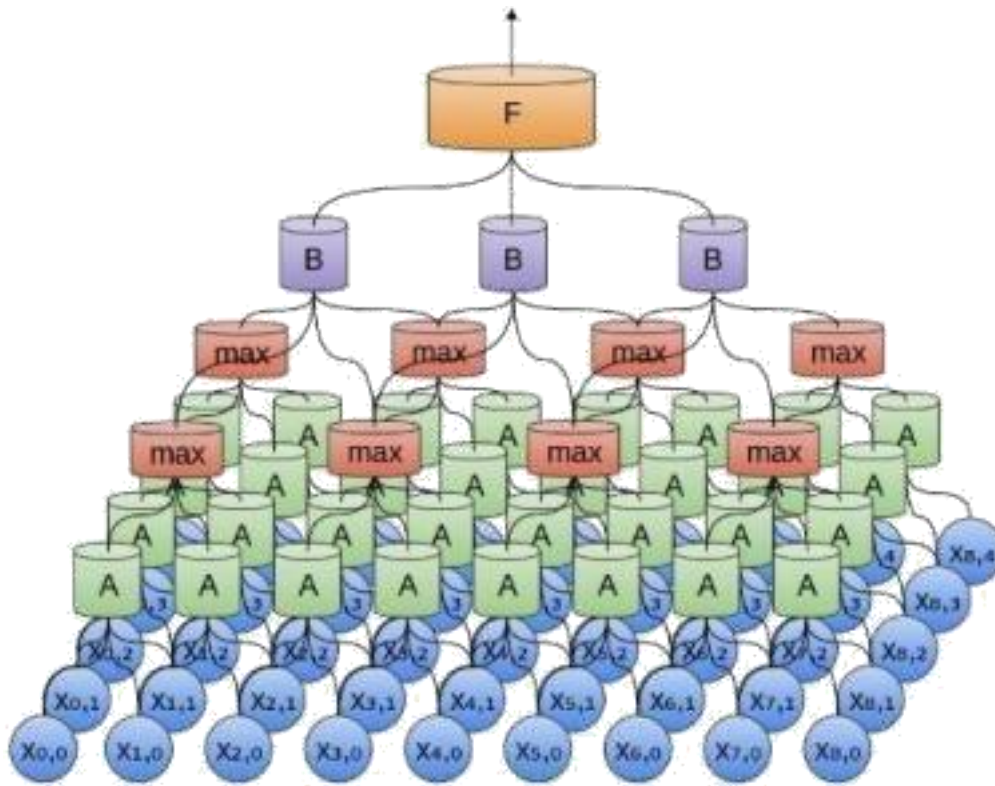


Figure 3.8: Complete 2D CNN

3.2 Recurrent Neural Networks (RNN)

Now, we need to relate the current input to the previous inputs in order to tell what exactly is happening in the video. Since, video is a time dependent collection of images, we need memory to store the images to evaluate the complete video. The major advantage of RNN is that it supports memory.

3.2.1 Theory of RNN (RNN)

In RNN or Recurrent Neural Network a directed graph is formed by connections between the nodes with a temporal sequence. Recurrent Neural Networks connects previous information to the current task. They might be used be used in cases of completing sentences and connecting current video frame to the previous frames. Recurrent Neural Networks have loops in them allowing the information to persist.

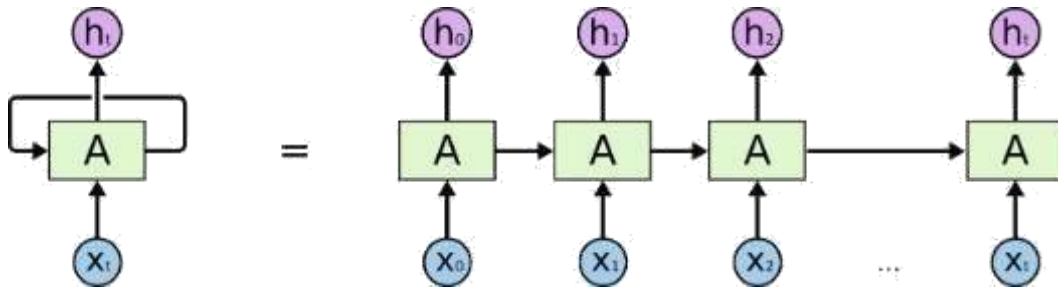


Figure 3.9: RNN structure

The output from an RNN is based on current input and previous outputs. Hence, it is a great approach when we need previous information.

3.2.2 Problem with a normal RNN

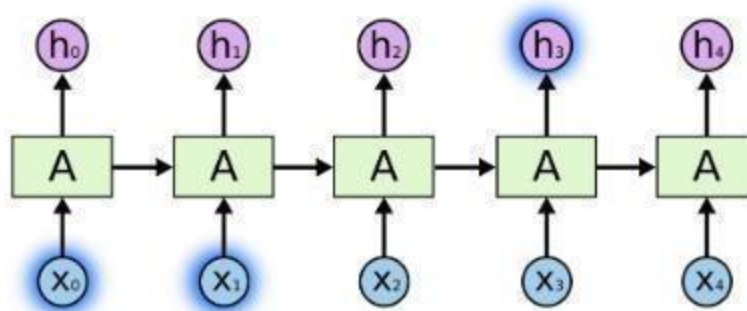


Figure 3.10: Information persisting in RNN

RNNs are useful in case when the gap between the current information and previous information is small. But, as the gap grows larger Recurrent Neural Networks aren't able to connect the point with the relevant information.

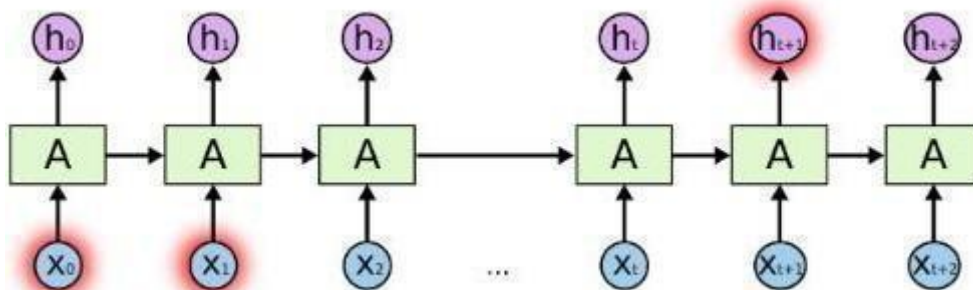


Figure 3.11: Loss in RNN

But in case of video, there is a possibility that the gap between the relevant content and the current point is very large. Hence, instead of using Recurrent Neural Networks, we use Long Short-Term Memory.

3.3 Long Short-Term Memory (LSTMs)

In video, we need a network that supports long term dependency. We need previous frames, to compute what is happening in the video, that is where LSTM comes into action.

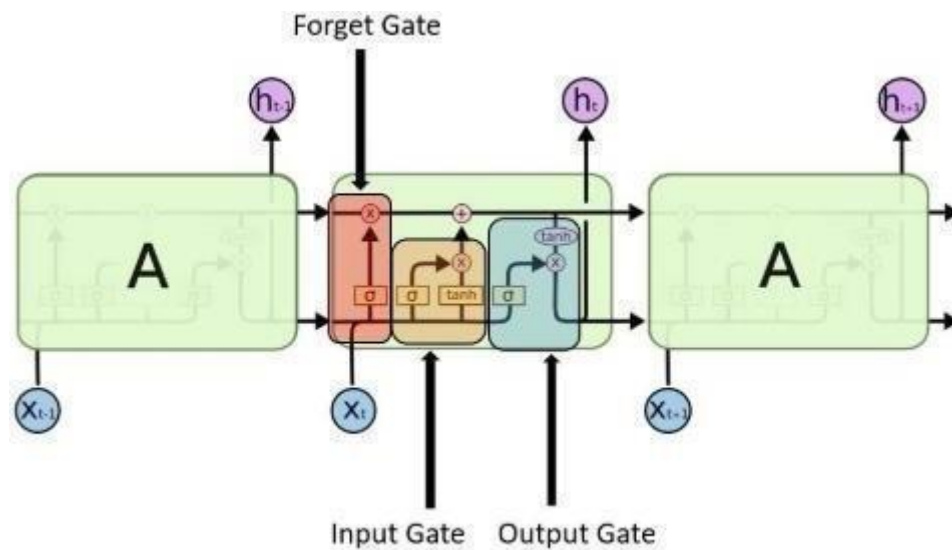
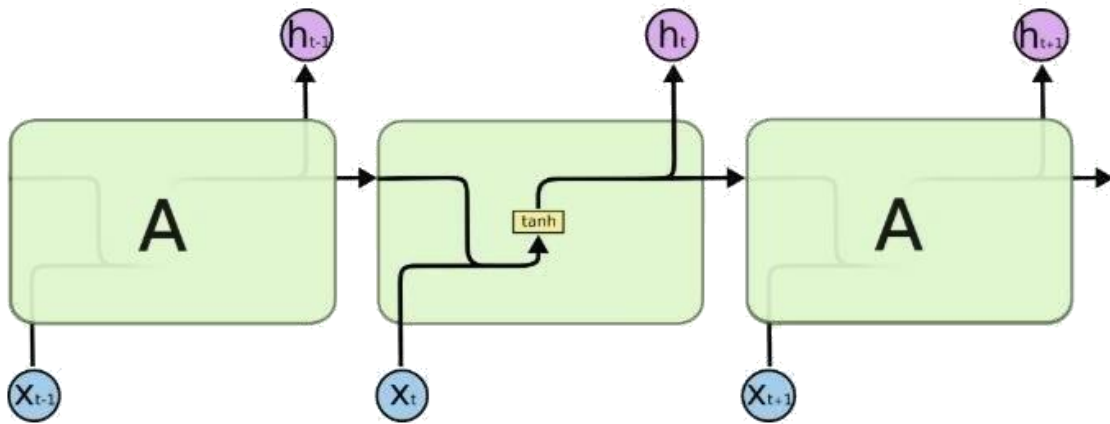


Figure 3.12: LSTM Structure

3.3.1 Theory of LSTMs

Long Short-Term Memory Networks or LSTMs are distinctive class of Recurrent Neural Networks that can learn long-term dependencies. These are designed explicitly to counter the problems faced in case of long-term dependencies.

Recurrent Neural Networks form chain of modules of neural networks repeating themselves. But, these RNNs they form a very basic structure (like tanh layer).



The repeating module in a standard RNN contains a single layer.

Figure 3.13: Repetitive structure of RNN

Long Short-Term Memory networks have quite similar chain like structure to those of regular RNNs, but the structure of repeating module is quite different. We have four neural network layers, instead of one neural network layer. These four layers interact in a quite different way.

3.3.2 Structure of LSTMs

The cell state runs like a straight line throughout the network with some linear interactions involved. It is quite possible that these interactions might not have a significant change on the next state.

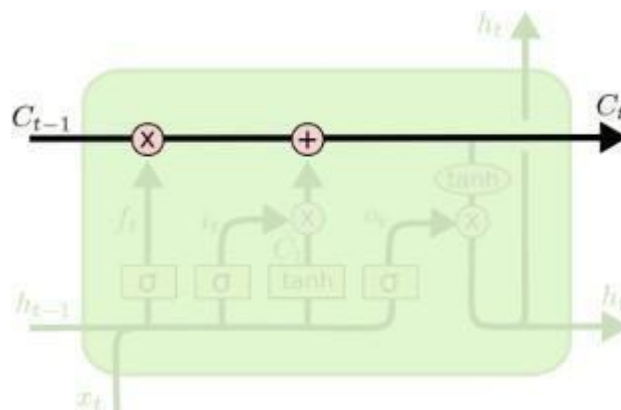


Figure 3.14: Linear cell state transformation

LSTMs involve certain structures that have the ability to add or delete information on the basis of relevance. They consist of pointwise multiplication operation and a sigmoid neural network layer.

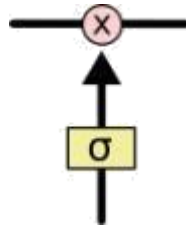


Fig 3.15: Gate

LSTMs consists of three gates:

- Input Gate

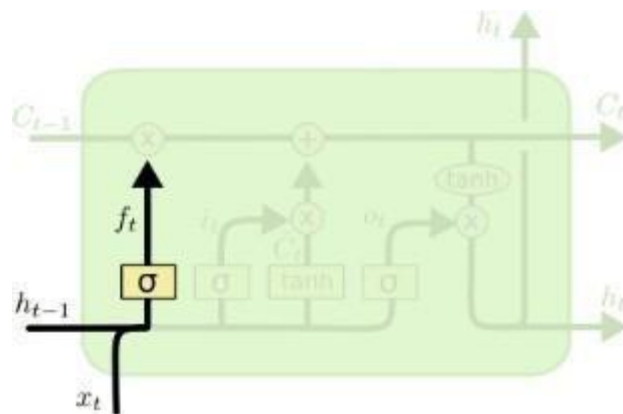


Fig 3.16: Input gate in LSTM

- Forget Gate

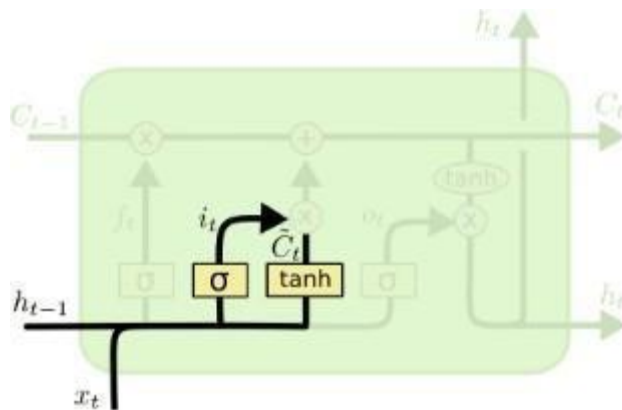


Fig 3.17: Forget Gate in LSTM

- Output Gate

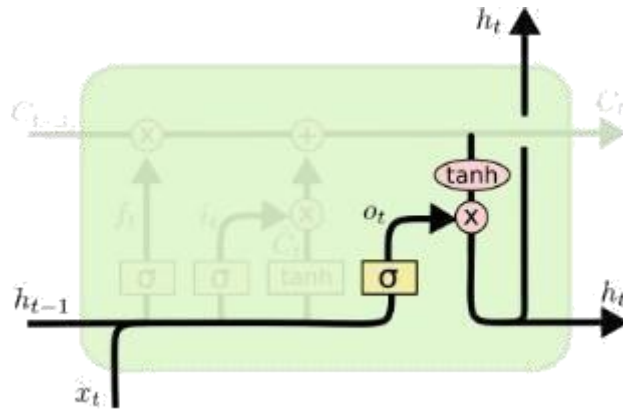


Fig 3.18: Output Gate in LSTM

3.3.3 Working of LSTMs

Firstly, we decide which information needs to be dumped away from the cell state. For this operation we use sigmoid layer which is also known as the “Forget Gate”. It uses previous state output h_{t-1} and input vector x_t and for every cell state C_{t-1} , it outputs a number between zero and one, where one means that we need to keep it and zero means that we have to forget it or dump it.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

In the second step, we decide that in the cell state which new information needs to be stored. For this, firstly we use sigmoid layer called the “input gate” that decides the values that need to be updated. Then, we create a vector of new candidate value using that can be added with the state using a tanh function. And finally, these two are combined to update the state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Then, we update old cell state with the new one. We then take the old state and multiply it by

f_t to forget the things that need to be forgotten and then we add $i_t * \tilde{C}_t$. This new candidate value is scaled by the amount we need to update the value.

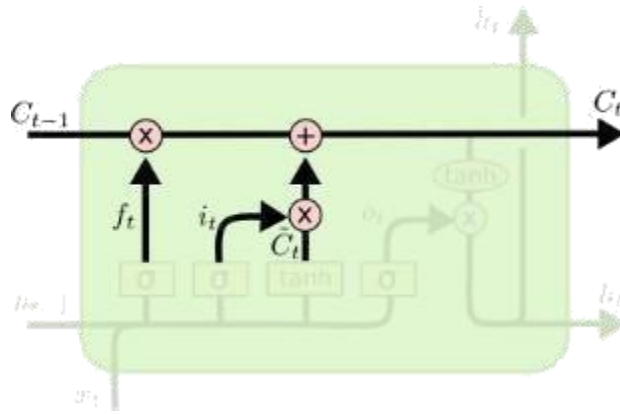


Fig 3.19: Addition of current input and previous state

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

At last, we decide the final output. Output will be a filtered version of our current cell state. Sigmoid layer decides what part of the cell state will be produced as the output, then it is multiplied with the cell state which is first passed through tanh. Hence, the output will only contain the parts that we wished for.

$$o_t = \sigma (W_o [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh (C_t)$$

3.4 Attention Mechanism

We use the complete image to find features every single time. This process takes a lot of time and is very inefficient as we only need to focus on certain regions and not the entire image.

3.4.1 Theory of Attention Mechanism

In order to focus only on the relevant regions, we use Attention Mechanism. It instead of focusing on the entire image, focus on certain regions that makes it faster to process the images and generate output.

There are two types of attention:

- Soft Attention

Soft Attention is a deterministic model that focuses on different regions and sub regions.



Fig 3.20: Soft Attention

- Hard Attention

Hard Attention is a probabilistic model that focuses on a single region.



Fig 3.21: Hard Attention

We make use of soft attention as we need to focus on multiple regions in order to evaluate what's happening.

3.4.2 Working of Attention Mechanism

Attention unit considers all sub regions as it's input and outputs weighted arithmetic mean of these regions. Arithmetic mean is the product of actual values and probabilities of those values.

$$E(X) = \sum_n p(X = X_n) X_n$$

The output from the CNN and context C are applied to weight and added and passed through tanh activation function to bring the value in the range of 0 to 1. Weights constitute the learnable parameters.

$$m_i = \tanh(y_i W_{y_i} + C W_C)$$

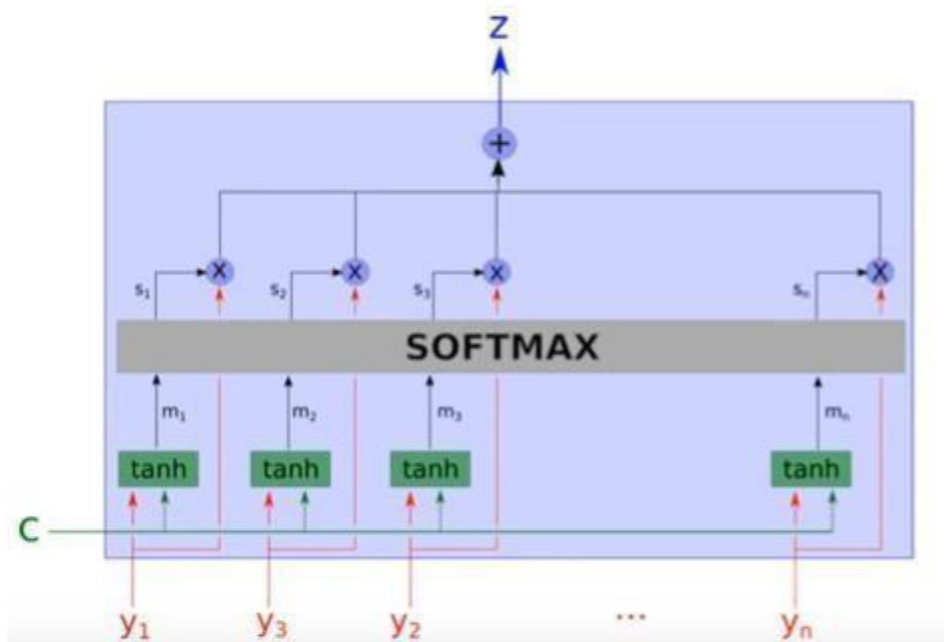


Fig 3.22: Working of attention mechanism

The similarity is given by the dot product of the output of CNN y and context c.

$$m_i = c y_i$$

Hence, similarity is directly proportional to the product. Therefore, the more relevant region will have more weight.

These m_i are passed through SOFTMAX function. This function outputs them as probabilities s_i .

$$s_i = \frac{e^{m_i}}{\sum_n e^{m_n}}$$

Finally, we take the inner product of the probability vectors s_i and sub regions y_i . We get output z_i of relevant regions of the entire image. Now, these outputs are passed as input to the LSTM modules.

4. PERFORMANCE ANALYSIS

4.1 Performance affected by Dataset.

The dataset used is made from MSR-VTT and MSVD dataset. It is a small dataset derived from these two huge datasets. Usage of a smaller subset and small number of epochs due to limited processing power available has altered the performance in a negative way. In spite of this, the model produces significantly good results.

4.2 Performance improvement using LSTM

Use of regular RNN may result in vanishing gradient problem or exploding gradient problem.

4.2.1 Vanishing Gradient

As we increase the number of layers in a Recurrent Neural Network, the gradient of the loss function approaches 0, which makes training of the data extremely difficult. The deeper the network, the greater the loss. This is because we use function like sigmoid function which brings down the input space between zero and one. Hence, its derivate becomes small and starts approaching zero. After a period, the model stops learning.

4.2.2 Exploding Gradient

In Recurrent Neural Networks, error gradients can result in a very large gradient value. This causes large updates to the weights in the network, which results in an unstable network. The values can become so large that they cause an overflow condition and produce NaN values.

In both the cases it becomes quite impossible to learn from the training data and the desired results aren't produced.

Use of LSTM eliminates both these problems. We found a significant improvement in our results when we used Long Short-Term Memory networks instead of a simple Recurrent Neural Network.

4.3 Performance improvement using Attention mechanism.

Using attention mechanism, the result was more precise as it focuses only on significant regions. Use of attention allowed us to use a much bigger dataset as it is significantly faster.

5. CONCLUSION

5.1 Conclusions

We have implemented attention based LSTM to caption videos. Use of this model produces an exceptional result as compared to other comparative models. This model not only increases accuracy, it is also efficient as it takes lesser time allowing us to work on bigger datasets and produce even better results.

5.2 Future Scope

A lot can be done in order to increase the efficiency of the model. The accuracy can be increased by increasing epochs. We can also use a bigger dataset for better results.

5.3 Applications

The proposed model can have a number of applications. It can be expanded further and used for detecting accidents. It can be used for automatic commentary generation in sports, it can be used for livestream in order to prevent illegal activities and avoid threat.

References

- [1]. Rakshith Shetty, Jorma Laaksonen “ Video captioning with recurrent networks based on frame and video level features and visual content classification”arXiv:1512.02949v1[cs.CL] 2015

- [2]. Yang Yang, Jie Zhou, Jiangbo Ai, Yi Bin, Alan Hanjalic, Heng Tao Shen , Yanli Ji “ Video Captioning by Adversarial LSTM ” arXiv:08410586 2018

- [3]. Subhashini Venugopalan, Marcus Rohrbach, Jeff Donahue, Raymond Mooney, Trevor Darrell, Kate Saenko “ Sequence to Sequence – Video to Text ”arXiv:1541.44544 2015

- [4]. Lianli Gao, Zhao Guo, Hanwang Zhang, Xing Xu, and Heng Tao Shen “ Video Captioning With Attention-Based LSTM and Semantic Consistency ”arXiv:07984828 2017

Appendices

Code for extracting frames:

```
import cv2

num = 7010
while (num <= 7509):
    vidcap = cv2.VideoCapture('./videos/video' + str(num) + '.mp4')
    success, image = vidcap.read()
    count = 0
    success = True
    while success and count <= 270:

        if (count == 0 or count == 54 or count == 108 or count == 162 or
            count == 216 or count == 270):
            cv2.imwrite("./frames1/frame" + str(num) + "%d.jpg" % count,
                image)# save frame as JPEG file
            success, image = vidcap.read()

print('Read a new frame: ', success)
count += 1
num += 1
```

Code for encoder:

```
class Encoder(tf.keras.Model):
    def __init__(self):
        super(Encoder, self).__init__()
    self.batch_sz = 64
    self.units = 256
    self.gru = tf.keras.layers.GRU(self.units,
        return_sequences = True,
        return_state = True,
        recurrent_initializer = 'glorot_uniform')
    def call(self, x, hidden):
        output, state = self.gru(x, initial_state = hidden)
        return output, state

    def initialize_hidden_state(self):
        return tf.zeros((self.batch_sz, self.units))
```

Code for decoder:

```
class RNN_Decoder(tf.keras.Model):
    def __init__(self, embedding_dim, units, vocab_size):
        super(RNN_Decoder, self).__init__()
        self.units = units

    self.embedding = tf.keras.layers.Embedding(vocab_size,
        embedding_dim)
    self.lstm = tf.keras.layers.LSTM(self.units,
        return_sequences = True,
        return_state = True,
        recurrent_initializer = 'glorot_uniform')
    self.fc1 = tf.keras.layers.Dense(self.units)
    self.fc2 = tf.keras.layers.Dense(vocab_size)

    self.attention = BahdanauAttention(self.units)

    def call(self, x, features, hidden): #defining attention as a
        separate model
        context_vector, attention_weights = self.attention(features,
            hidden)# x shape after passing through embedding ==
            (batch_size, 1, embedding_dim)
        x = self.embedding(x)

        # x shape after concatenation == (batch_size, 1, embedding_dim
            hidden_size)
        x = tf.concat([tf.expand_dims(context_vector, 1), x], axis = -1)

        # passing the concatenated vector to the LSTM
        output, state, _ = self.lstm(x)
        # shape == (batch_size, max_length, hidden_size)
        x = self.fc1(output)

        # x shape == (batch_size * max_length, hidden_size)
        x = tf.reshape(x, (-1, x.shape[2]))

        # output shape == (batch_size * max_length, vocab)
        x = self.fc2(x)

    return x, state, attention_weights

    def reset_state(self, batch_size):
        return tf.zeros((batch_size, self.units))
```

Code for attention:

```
class BahdanauAttention(tf.keras.Model):
    def __init__(self, units):
        super(BahdanauAttention, self).__init__()
        self.W1 = tf.keras.layers.Dense(units)
        self.W2 = tf.keras.layers.Dense(units)
        self.V = tf.keras.layers.Dense(1)

    def call(self, features, hidden): #features(CNN_encoder output)
        shape == (batch_size, 64, embedding_dim)

        # hidden shape == (batch_size, hidden_size)# hidden_with_time_axis
        shape == (batch_size, 1, hidden_size)
        hidden_with_time_axis = tf.expand_dims(hidden, 1)

        # score shape == (batch_size, 64, hidden_size)
        score = tf.nn.tanh(self.W1(features) + self.W2
            (hidden_with_time_axis))

        # attention_weights shape == (batch_size, 64, 1)# you get 1 at the
        # last axis because you are applying score to self.V
        attention_weights = tf.nn.softmax(self.V(score), axis = 1)

        # context_vector shape after sum == (batch_size, hidden_size)
        context_vector = attention_weights * features
        context_vector = tf.reduce_sum(context_vector, axis = 1)

    return context_vector, attention_weights
```

Code for loss function:

```
def loss_function(real, pred):
    mask = tf.math.logical_not(tf.math.equal(real, 0))
    loss_ = loss_object(real, pred)
    mask = tf.cast(mask, dtype = loss_.dtype)
    loss_ *= mask
    return tf.reduce_mean(loss_)

checkpoint_path = "./checkpoints/train"
ckpt = tf.train.Checkpoint(encoder = encoder,
    decoder = decoder,
    optimizer = optimizer)
ckpt_manager = tf.train.CheckpointManager(ckpt, checkpoint_path,
    max_to_keep = 5)

start_epoch = 0
if ckpt_manager.latest_checkpoint:
    start_epoch = int(ckpt_manager.latest_checkpoint.split('-')[-1])#
    restoring the latest checkpoint in checkpoint_path
ckpt.restore(ckpt_manager.latest_checkpoint)
```

ORIGINALITY REPORT

8%

SIMILARITY INDEX

4%

INTERNET SOURCES

7%

PUBLICATIONS

5%

STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

4%

★ Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, Kate Saenko. "Sequence to Sequence -- Video to Text", 2015 IEEE International Conference on Computer Vision (ICCV), 2015

Publication

Exclude quotes On

Exclude bibliography On

Exclude matches < 10 words

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 14-07-2020



Type of Document (Tick): PhD Thesis M.Tech Dissertation/ Report B.Tech Project Report Paper

Name: Himanshu Gupta Department: IT Enrolment No 161454

Contact No. 9906906682 E-mail. himu.gupta23@gmail.com

Name of the Supervisor: Dr. Suman Saha

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): Video Captioning Using LSTM with Attention Mechanism

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/ revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages = 41
- Total No. of Preliminary pages = 8
- Total No. of pages accommodate bibliography/references = 4

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at⁸.....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor)

Signature of HOD

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	<ul style="list-style-type: none"> • All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String 		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature

Librarian

.....

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 14-07-2020



Type of Document (Tick): PhD Thesis M.Tech Dissertation/ Report B.Tech Project Report Paper

Name: Shubham Garg Department: IT Enrolment No 161456

Contact No. 7018445885 E-mail. sh.shubham.garg@gmail.com

Name of the Supervisor: Dr. Suman Saha

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): Video Captioning Using LSTM with Attention Mechanism

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/ revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages = 41
- Total No. of Preliminary pages = 8
- Total No. of pages accommodate bibliography/references = 4

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at⁸.....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor)

Signature of HOD

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	<ul style="list-style-type: none"> • All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String 		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature

Librarian

.....

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com