Video Captioning using First and Last frame

A Project Report

submitted by

VISHAL B M

in partial fulfilment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY AND MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

MAY 2019

THESIS CERTIFICATE

This is to certify that the thesis titled **Video Captioning using First and Last frame**, submitted by **Vishal B M**, to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor of Technology and Master of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. A.N. Rajagopalan
Research Guide
Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: 5th May 2019

ACKNOWLEDGEMENTS

I would like to thank Prof. A. N. Rajagopalan for being an excellent mentor and a teacher. I am very grateful for being a part of IPCV lab since it has a great research atmosphere. This project has given me a chance to experience a significant side of academic research. I am sure it will be helpful in my further endeavours.

I would like to thank IIT Madras for giving this great opportunity. It was a good learning experience to have worked with research scholars in IPCV lab. It would have been difficult without the help of Maitreya, Praveen and Mahesh. I would also like to thank rest of the labmates for including me in their discussions and making the lab a fun place to work.

Finally, I would like to thank my family and friends who have been supportive all along.

ABSTRACT

KEYWORDS: captioning, summarization, MSVD, Teacher-Student network, BLEU

Over the past few years, video understanding has received a lot of attention. It can be captioning, classification or summarization that requires understanding subjects and actions in the video. Although the problem statement may seem very abstract, there have been quite a few works in the past four years. Here, Video captioning has been discussed in detail.

One of the problem to be tackled in video captioning is summarizing multiple shots. Mixing of video features from multiple shots will lead to a sub-par caption. One way to tackle the problem is to separate one shot from another using a shot detector. Another problem with video captioning is the length of the video. If the video is too long, it can be difficult to understand everything happening in the video. Solution to this problem is to use constant number of frames irrespective of the length of the video and train the network to generate same caption. So, the network should be capable to "interpolate" the captions from missing frames.

A relatively new paradigm called Teacher-Student networks is used. MSVD and MSR-VTT are the two datasets used here. There is an improvement of approximately 3% in BLEU score of generated captions using teacher-student network compared to using the original network.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long short term memory
BPTT	Backpropagation through time
BLEU	Bilingual evaluation understudy
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
CIDEr	Consensus-Based Image Description Evaluation
METEOR	Metric for Evaluation of Translation with Explicit Ordering
MSVD	Microsoft Video Description(dataset)
MSR-VTT	Microsoft research Video to text
SCN	Semantic compositional network

CHAPTER 1

INTRODUCTION

Describing a video using natural language, called Video Captioning, has been an important area of research in recent years. This is a crucial part of machine intelligence and also has a number of potential applications. The problem is to generate a semantically meaningful sentence that appropriately describes contents and actions taking place in a video. Since the problem is more abstract than say deblurring, researchers tend to apply deep learning algorithms.

Image Captioning, describing an image using natural language, also had a recent surge of interest. Image captioning takes a single image whereas video captioning takes multiple images(or Frames) for generating a caption. So, the interest lies in generating as good a caption generated by using all the frames but by only using two frames.

This thesis revolves around video captioning using only first and last frame. It has been divided into 4 parts. First part deals with basic concepts used in this thesis. Concepts such as Convolutional Neural Network, Recurrent Neural Network, Long short Term Memory network and some early video captioning networks are discussed. There is an introduction to word embeddings and how a sentence is created using sequence modelling networks.

The next part deals with datasets used in the project and various evaluation metrics commonly used for captioning. The following part presents main ideas of 2 research papers. Namely, Hierarchical boundary aware neural encoding and Semantic Compositional Networks. The last part showcases results on the above mentioned datasets and further directions.

CHAPTER 2

Relevant concepts

Following sections will help in understanding various ideas used in this thesis. CNNs are used to extract relevant features from images and videos. A brief introduction of RNNs and how they are used in sequence modelling. Here, LSTMs were used instead of RNNs to increase the networks capacity to "remember" features.

There is an introduction to word embeddings and sentence formation to familiarize with sequence modelling. Formation of a sentence is based on maximizing the probability of words occurring together. Beam search is one such algorithm used in language models. A brief introduction to attention based sequence modelling is also given. There will be certain frames in a video that contribute more for the caption than others. So, giving more attention to that will improve surely improve the quality of captions generated.

2.1 Convolutional Neural Networks

Convolutional Neural Networks(CNNs) introduced by LeCun *et al.* (1999) have been very successful in encoding or decoding relevant features from images. Since it is a deep learning algorithm, there is a scope for using the same network for variety of problems. Training with the same network and changing only the loss function will still be able to beat traditional methods by a considerable margin. In recent years, CNNs are mainly used in image segmentation, enhancement, classification and captioning.

Local connectivity and parameter sharing are the two important points in favour of CNNs. Local connectivity is a sensible assumption that features around a pixel is affected only by pixels attached locally to it. Two pixels far away cannot affect each other's features. Parameter sharing reduces number of parameters to be learnt since dimensions of an image can be arbitrarily large. Also parameter sharing assumes that the user is looking for similar features repeated in different areas in an image.

After a convolution layer, pooling of features is carried out. Pooling is an operation used for reducing the size of feature maps so that the dimension of MLP in the end is not too large. Two main kinds of pooling are Max pooling and average pooling. These

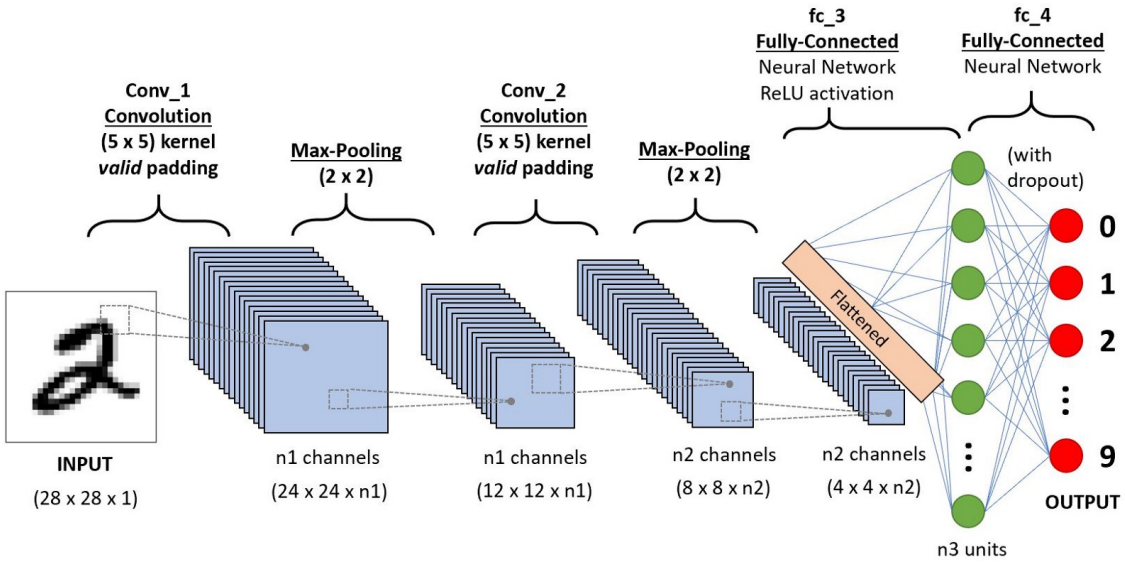

Figure 2.1: An example of Convolutional Neural Network

layers are followed typically by non-linearities like ReLU, Sigmoid, TanH.

There are a number of pretrained image classification CNN networks trained on Imagenet Dataset of 2M images by Deng *et al.* (2009). It is safe to assume that these networks extract distinct enough features to distinguish each image among 1000 classes. ResNet-152 introduced by He *et al.* (2016) is one such network that is used in this thesis for extracting features from videos.

2.2 Recurrent Neural Networks

Recurrent Neural Network(RNN) introduced by Rumelhart *et al.* (1986) is used if the data to be encoded or decoded is sequential. The most common sequential data is a natural language sentence. RNNs have been successful in solving sequence to sequence problems. Converting a sentence in one language to a meaningful sentence in another language is one such problem.

Referring to the figure above, x_t is the input, o_t is the output and s_t is the intermediate output at time t. Intermediate output s_t serves as a memory unit for RNNs. Since in a sequential data, unlike CNNs, there can be global connectivity in data(only local connectivity in CNNs), there is a necessity to carry features along with time. Both data(x_t)

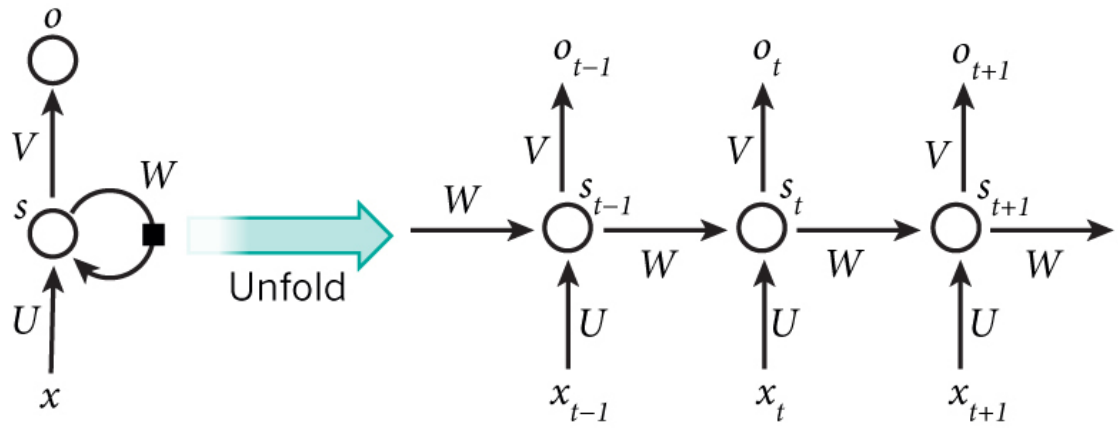

Figure 2.2: An example of Recurrent Neural Network

and intermediate output(h_{t-1}) will be inputs for the next RNN unit to produce o_t. Back Propagation Through Time(BPTT) is used for training RNNs.

Main problem with RNNs is vanishing and exploding gradients with increasing sequence length during BPTT. RNNs should keep memories to capture long distance relationships in a sequence. Below are the equations that govern a Vanilla RNN. U,V,W are learnt weights during BPTT.

$$s_t = tanh(Ws_{t-1} + Ux_t + b_1)$$
$$o_t = softmax(Vs_t + b_2)$$

2.3 Long Short-Term Memory

LSTM introduced by Hochreiter and Schmidhuber (1997) is another type of RNN that selectively reads, writes and forgets depending on the current input. For example, there are some words in a normal english sentence such as "an","the","a" that do not exactly contribute any special meaning to the sentence. These words will be forgotten along the way in LSTM network. This network can capture memories in a long sequence. Below are the gate equations governing a single LSTM cell.

$$o_t = \sigma(W_o s_{t-1} + U_o x_t + b_o)$$
$$i_t = \sigma(W_i s_{t-1} + U_i x_t + b_i)$$
$$f_t = \sigma(W_f s_{t-1} + U_f x_t + b_f)$$

Below are the equations that hold states s_t and memory c_t

$$c_t^{'} = \sigma(W c_{t-1} + U x t + b)$$
$$c_t = f_t \circ c_t + i_t \circ c_t^{'}$$
$$s_t = o_t \circ tanh(c_t)$$

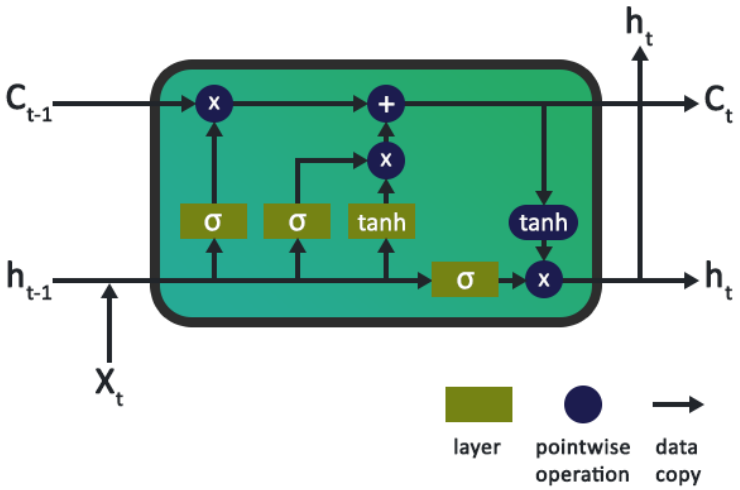

Figure 2.3: An example of Long Short Term Memory cell

2.4 Video Captioning networks

Focussing on recent deep learning based video captioning networks, one of the first works to use recurrent neural networks was introduced by Venugopalan *et al.* (2014) Video was represented using single frame CNN extracted features mean pooled together to form one feature map. The feature map will be then fed to a LSTM network to decode the sentence. The drawback in this case is that mean pooling is independent of the sequence.

Donahue *et al.* (2014) overcame this drawback by encoding the sequence by using another LSTM layer instead of a CNN. Then this feature map will be decoded using a Conditional Random Field to get semantic tuples of activity, object and location. Then another LSTM layer to form a sentence using these semantic tuples.

Venugopalan *et al.* (2015) came up with a complete neural network architecture for both encoding and decoding parts of the network. They used a stacked LSTM to read

video features and another stacked LSTM conditioned on the previous LSTM for decoding captions. Other works have followed this kind of approach. Other important works such as Yao *et al.* (2015) propose a temporal attention mechanism that allows to go beyond local temporal modeling and learns to automatically select the most relevant temporal segments given the text-generating RNN.

Pan *et al.* (2015*b*) Simultaneously explored the learning of LSTM and visual-semantic embedding. The former aims to locally maximize the probability of generating the next word given previous words and visual content, while the latter is to create a visual-semantic embedding space for enforcing the relationship between the semantics of the entire sentence and visual content.

Venugopalan *et al.* (2016) improved decoding with the help of a large text corpora. Rohrbach *et al.* (2015) tries to learn semantic tuples consisiting of verbs, objects and places separately using different networks and putting them all together to form meaningful sentence.

Pan *et al.* (2015*a*) improved the video encoder by proposing Hierarchical Recurrent Neural Encoder(HRNE). A second layer of LSTM is introduced to reduce the number of LSTMs a video representation has to go through before embedded in the output. In this way, the memory burden on LSTM will be decreased to perform better.

Yu *et al.* (2015) proposed Video paragraph captioning that produces one simple short sentence that describes a specific short video interval. It exploits both temporal- and spatial-attention mechanisms to selectively focus on visual elements during generation. The paragraph generator captures the inter-sentence dependency by taking as input the embedding produced by the sentence generator, combining it with the paragraph history, and outputting the new initial state for the sentence generator.

2.5 Word embeddings and sentence formation

Word embeddings introduced by Mikolov *et al.* (2013) are used in language modelling for predicting the "next word" in a sentence. A typical language model will be

based on a vocabulary of size, say 10,000. Each word can be represented as a one-hot vector. The problem with one-hot vector is that there cannot be any transfer learning at a word level since dot product of any two one-hot vector is zero.

Let's say there is a sentence that ends with "orange juice". If someone wants to predict the next word of "Apple", unless there is a similarity between orange and apple, it will be difficult to predict the next word as "juice". So, word embeddings generally come with a relatively lower dimensions, for example, 300. Each of the 10,000 word will have a vector of dimension 300 assigned in the word embedding. Generally word embeddings are packaged with the captioning dataset.

Generally if there is a word in a sentence not present in the vocabulary, it will be represented as "<UNK>" token. Also, there is a token for end of sentence represented as "<EOS>". Sentence will be generated till the end of sentence token is generated. While Training, sentences are appended with "<EOS>" token for learning the pattern of words used to learn end of sentences.

Word embeddings depend on the vocabulary that is used for training it. The most common algorithm used for creating an embedding is called Word2Vec by Tomas Mikolov et.al.. A multilayer perceptron with softmax activation is used to train the word embedding. Each word in the vocabulary is assigned an index. Word2Index and Index2Word are two dictionaries also packed into the captioning dataset. For example the following sentence-"a man is walking on the road" may be encoded as [9,103,1143,4,1,2897]. Each index represents the corresponding column in the word embedding.

There is a slight difference between training and testing a RNN as supposed to training and testing a CNN. While training, input sequence is given one by one after the other to RNN. Output loss is back propagated through time for every RNN unit. While testing, output of a RNN unit is given as input to the next RNN unit. The output of RNN units is a softmax operation to decide the next word given previous words.

2.6 Beam search

Beam search in sequence to sequence learning was used by Wiseman and Rush (2016). Sentence formation can be achieved by taking softmax at every RNN/LSTM unit and still get a meaningful sentence. But, there might be a better sentence possible that was not considered at all. Taking softmax and considering the best word each time is a type of greedy search. Greedy search may not work all the time. Beam search on the other hand considers a number of sentences at a time and considers the sentence whose joint probability is in the top B sentences. B is a hyperparameter and is generally less than 10.

2.7 Attention based sequence modelling

Attention based modelling was first introduced by Bahdanau *et al.* (2015) in his paper regarding neural machine translation. It is observed that translation of long sequences generally taken part by part otherwise, BLEU score keeps falling as the length of the sequence grows.

The idea of attention is that a word in the output sequence is going to depend on only a few or less words in the input sequence. In the figure below, it can be seen that a linear combination of features with coefficients as $\alpha(\cdot, \cdot)$ is used as the context.

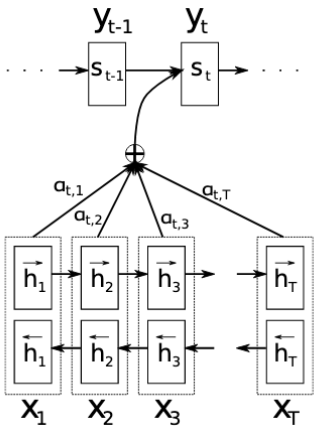

Figure 2.4: Attention based sequence modelling

For predicting a word at time t, context may depend solely on a single word (which happens in case of nouns). In that case α will be close to one for features of that particular word and zero for rest of the features since sum of $\alpha(t,:)$ should be 1. These attention weights are learnt during back propagation. It has been shown that attention based sequence modelling performs well in case of long sequences.

This idea can be carried over to video captioning. Instead of attention weights for words, attention can be paid to certain frames more than the others.

CHAPTER 3

Evaluation Metrics and Datasets

In the previous chapter, types of networks that are used in this thesis were introduced. The use of LSTMs in captioning is ubiquitous. This chapter revolves around metrics and datasets used in video captioning. Evaluating if two sentences are same can be difficult. It can happen that the meaning of two sentences are same but the words are jumbled. There are several evaluation metrics that are used in the literature. Some of the important metrics being BLEU, METEOR and ROUGE.

3.1 BLEU

Bilingual Evaluation Understudy introduced by Papineni *et al.* (2002) is one of the most popular evaluation criteria in Natural Language Processing. BLEU score is often calculated as the geometric mean of 4 BLEU-n scores(n being from 1 to 4). BLEU-i is calculated by dividing the number of times an i-words subsequence in generated sentence occurs in reference sentence by number of times it occurs in generated sentence. BLEU is always between 0 and 1.

$$\text{BLEU} = \min(1, \frac{output-length}{reference-length})(\prod_{i=1}^{i=4} BLEU - i)^{\frac{1}{4}}$$

If the generated sentence is very short, it will easy to get high BLEU score even if the sentence is not good enough. Therefore there is a brevity penalty multiplied to geometric mean to discourage generation of short sentences.

3.2 METEOR

METEOR introduced by Lavie and Agarwal (2007)is another metric for machine translation evaluation, and it claims to have better correlation with human judgement. We try to find the largest subset of mappings that can form an alignment between the candidate and reference translations. For this, we look at exact matches, followed by matches after Porter stemming, and finally using WordNet synonymy. After such an

alignment is found, suppose m is the number of mapped unigrams between the two texts. Then, precision and recall are given as $\frac{m}{c}$ and $\frac{m}{r}$, where c and r are candidate and reference lengths, respectively. F is calculated as,

$$F = \frac{PR}{(\alpha P + (1-\alpha)R)}$$

To account for the word order in the candidate, penalty function is introduced.

$$P = \gamma(\frac{c}{m})^{\beta}$$

Here, c is the number of matching chunks and m is the total number of matches. As such, if most of the matches are contiguous, the number of chunks is lower and the penalty decreases. Finally, the METEOR score is calculated as $(1 - Penalty)F$

3.3 ROUGE

ROUGE was introduced by Lin (2004)There are various types of ROUGE score. ROUGE-N/L/W/S are types commonly seen. ROUGE-N will be explained here. This is based on n-grams. For example, ROUGE-1 counts recall based on matching unigrams, and so on. For any n, we count the total number of n-grams across all the reference summaries, and find out how many of them are present in the candidate summary. This fraction is the required metric value.
Suppose A and B are candidate and reference summaries of lengths m and n respectively. Then, we have

$$P = \frac{LCS(A,B)}{m}, R = \frac{LCS(A,B)}{n}$$

Where LCS is Longest Common Subsequence. That can be calculated efficiently using dynamic programming. F is calculated as,

$$F = \frac{(1+b^2)PR}{R+b^2P}$$

Here, b is a hyperparameter.

3.4 Datasets

3.4.1 MSVD

MicroSoft Video Description corpus is a set of 1970 youtube videos with multilingual captions. Average length of the videos is 10 seconds with an average of 8 words in English captions. Following are the example videos and reference captions.

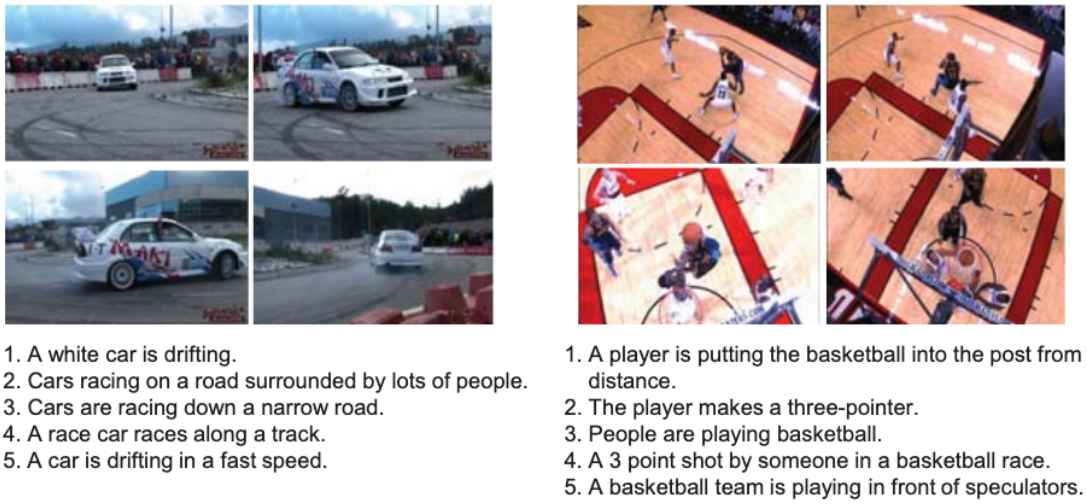

Figure 3.1: Example videos and captions in MSVD dataset

3.4.2 MSR-VTT

MicroSoft Research Video To Text dataset presented by Xu *et al.* (2016)is a video captioning dataset with 10,000 videos of generic day-to-day activity. Average duration of the videos being 20 seconds and average number of words is 10. These video clips last for 41.2 hours in total, covering the 20 representative categories and diverse visual content collected with 257 queries in the video engines. The dataset contains 200K clip-sentence pairs, and each clip is annotated with about 20 natural sentences. Compared to MSVD, MSR-VTT is more challenging, due to the large variety of videos.

3.4.3 Problems associated with the above datasets

The above datasets provide a variety of videos for captioning. The problem we are trying to solve is to generate captions given only first and last frame. The above two datasets which we have used are not a cause-effect type videos. Repetitive tasks in the

1. A black and white horse runs around.
2. A horse galloping through an open field.
3. A horse is running around in green lush grass.
4. There is a horse running on the grassland.
5. A horse is riding in the grass.

1. A woman giving speech on news channel.
2. Hillary Clinton gives a speech.
3. Hillary Clinton is making a speech at the conference of mayors.
4. A woman is giving a speech on stage.
5. A lady speak some news on TV.

Figure 3.2: Example videos and captions in MSR-VTT dataset

video rendering video captioning unnecessary. Image captioning can be used instead to find better captions.

Figure 3.3: An arbitrary first and last frame with respect to the main activity i.e, water skiing

Since we are only going to use features of first and last frame, a video will be useless if either first and/or last frame is arbitrary or completely different from the subject of the video. Many of the datasets only contain a single sentence for the whole video although it can be described better with a paragraph. There are no video to paragraph with cause-effect type video datasets currently accessible to public.

CHAPTER 4

Relevant Video captioning models

In this chapter, relevant video captioning papers are discussed. Specifically, the teacher networks that are tried for Teacher-Student paradigm are examined. Loss function and network architecture used in each paper is discussed. Results of captioning MSVD and MSR-VTT datasets using 2 networks are compared and best of them is selected as Teacher network.

4.1 Teacher-Student network

Video captioning using only first and last frame can be interpreted as captioning with limited information. There is a class of algorithm that deals with limited information or limited network complexity. Those algorithms come under Teacher-Student paradigm.

Bhardwaj and Khapra (2018) focuses on the task of video classification and aim to reduce the computational time by using the idea of neural network distillation. Specifically, first train a teacher network which looks at all the frames in a video and computes a representation for the video. Then train a student network whose objective is to process only a small fraction of the frames in the video and still produce a representation which is very close to the representation computed by the teacher network. This smaller student network involving fewer computations can then be employed at inference time for video classification.

The student network takes every j^{th} frame as the input. Video classification and video captioning both have the same cross-entropy loss. Therefore, the same architecture can be used for video captioning while feeding features of only first and last frame.

$$L_{teacher} = -\sum_{i=1}^{C} y_i \log y_i^{'}$$
$$L_{student} = -\sum_{i=1}^{C} y_i^{'} \log y_i^{"} + \lambda \mid I_T - I_S \mid^2$$

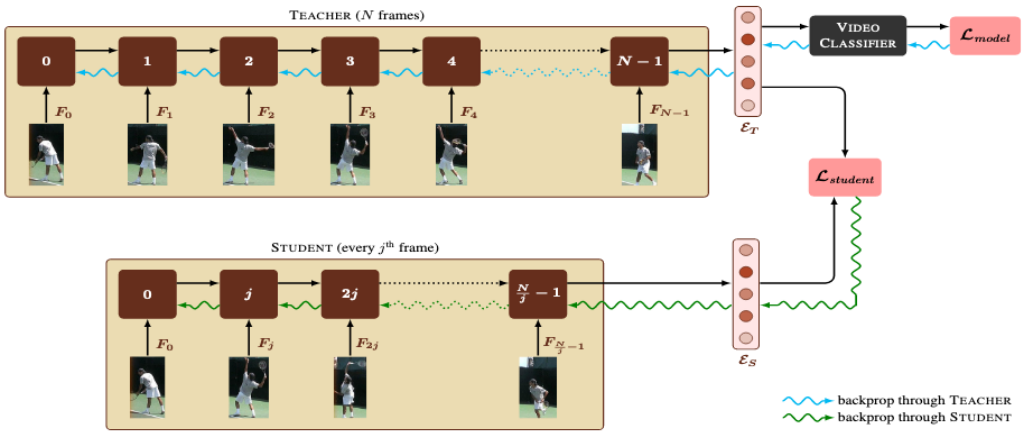

Figure 4.1: The Teacher-Student Video classification model

Teacher-Student network needs a teacher network that is able to generate captions with high BLEU score since student network is trying to have same features as that from the teacher network. The loss function for training such a network can be a cross entropy loss between representation computed by teacher network and representation computed by student network.

Loss in student network is computed by taking $y_i^{'}$, softmax output of parent network as the true label and $y_i^{"}$, softmax output of student network as the predicted label. Addition to that there is another loss between the intermediate representation computed by both networks.
The teacher network can be any of the state of the art models. Following are such models chosen for Teacher network. Details of experiments on those models are as follows.

4.2 Sequence to Sequence- Video to Text

This paper presented by Venugopalan *et al.* (2015) is the first to use LSTMs for both encoding video and decoding captions. Video captioning is analogous to machine translation between natural languages, where a sequence of words in the input language is translated to a sequence of words in the output language. The main idea to handle variable-length input and output is to first encode the input sequence of frames, one at a time,representing the video using a latent vector representation,and then decode from that representation to a sentence, one word at a time.

This is an important model that was followed by all other models for video captioning.

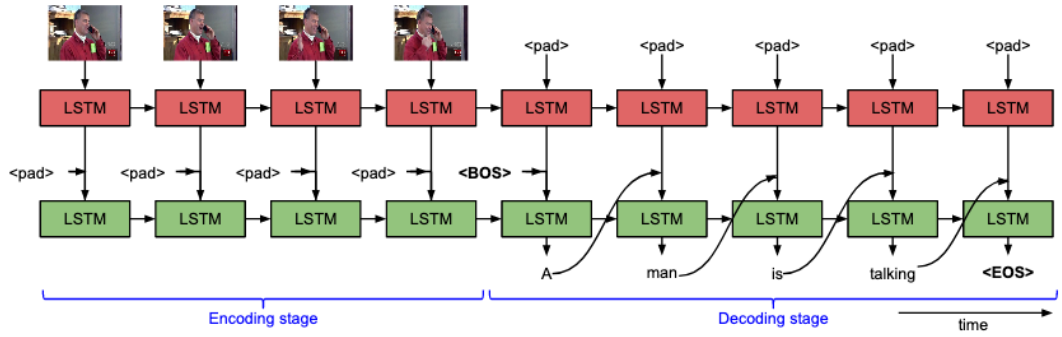

Figure 4.2: Sequence to Sequence- Video to Text model

The token "<pad>" is a zero input since some input should be given to LSTM. Encoding stage extracts relevant features for captioning and passes on to the decoding stage. The latent feature vector that is passed from encoding stage to decoding stage is known as the "context". A token "<BOS>" or beginning of sentence is given by the user to indicate that captioning should begin after this word.

4.2.1 Video and Text feature representation

Each Video frame is passed through a pretrained CNN model. Here, a variant of AlexNet and also VGG16 is used. Each frame is scaled to 256x256 and cropped to 227x227 before feeding it to the above CNN models. The result is a 500 dimension linear embedding formed at the last but one layer of the CNNs.

In addition to CNN outputs from RGB frames, optical flow is also incorporated that measures as input sequences to the architecture. Many papers have shown that incorporating optical flow information to LSTMs improves activity classification. As many of the descriptions are activity centered, it is bound to improve the captioning task as well.

Text input is one-hot encoded and converted to a lower dimension of 500 using another neural network. Word embedding were not used in this case. SO this embedded word vector is concatenated with feature vector h_t created by the first LSTM layer is sent to the second LSTM.

4.2.2 Training

While training, the ground truth will be given to the LSTMs as an input and loss is calculated with respect to the ground truth. While testing, words generated in current LSTM unit is given to the subsequent LSTM as the input. While training in the decoding stage, the model maximizes for the log-likelihood of the predicted output sentence given the hidden representation of the visual frame sequence, and the previous words it has seen.

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{t=1}^{m} \log p(y_t | h_{n+t-1}, y_{t-1}; \theta)$$

During the decoding stage, the most possible word is selected using softmax and not using beam search.

$$p(y|z_t) = \frac{\exp W_y z_t}{\sum_{y' \epsilon V} \exp W_{y'} z_t}$$

4.3 Hierarchical Boundary-Aware Video Captioning

Proposed by Baraldi *et al.* (2017) the core idea of this network is to prevent mix-up of memory passed by LSTMs if there is a shot change. So, if there is a shot detector, Summarizing the features of one shot and passing that to a new LSTM will prevent the mix-up of those features. If there is a shot change, there will be reinitialization of state and memory in old LSTM.

Given an input video, there is a recurrent video encoder which takes as input a sequence of visual features($x_1, x_2, ..., x_n$) and outputs a sequence of vectors($s_1, s_2, ..., s_m$) as the representation for the whole video. The following figure only depicts the encoder part. Decoder consists of traditional LSTM network generating the sentence one word at a time.

Time boundary-aware recurrent cell is built on top of a LSTM unit. Update operations on the memory cell are modulated by three gates i_t, f_t and o_t, which are all computed as a combination of the current input x_t and of the previous hidden state h_{t-1}, followed by a sigmoid activation.

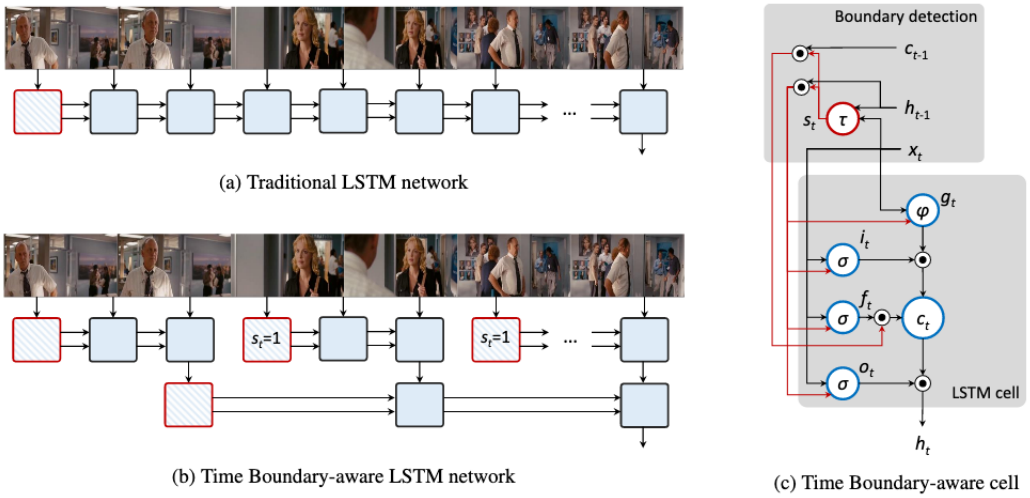

Figure 4.3: Difference between a traditional LSTM and Boundary aware LSTM network

At each time step, we select whether to transfer the hidden state and memory cell content to the next time step or to reinitialize them, interrupting the seamless update and processing of the input sequence. This depends on a time boundary detection unit, which allows our encoder to independently process variable length chunks of the input video.The boundaries of each chunk are given by a learnable function which depends on the input, and are not set in advance.

Formally, the boundary detector $s_t \epsilon \{0, 1\}$ is computed as a linear combination of the current input and of the hidden state, followed by a function f which is the composition of a sigmoid and a step function:

$$s_t = f(V_s^T \cdot (W_{si} x_t + W_{sh} h_{t-1} + b_s)$$

$$f(x) = \begin{cases} 0 & \sigma(x) \leq 0.5 \\ 1 & otherwise \end{cases}$$

where v_s^T is a learnable row vector and W_{sh},b_s are learned weights and biases. Given the current boundary detection s_t, before applying the memory unit update equations, the following substitutions are applied to transfer or reinitialize the network hidden state and memory cell at the beginning of a new segment, according to s_t:

$$h_{t-1} = h_{t-1} \cdot (1 - s_t)$$

$$c_{t-1} = c_{t-1} \cdot (1 - s_t)$$

The resulting state and memory are now employed to recompute the gates values, which will in turn be used for advancing to the next time step. The encoder produces an output only at the end of a segment. Ifs_t= 1, the hidden state of time step tâĹŠ1 is passed to the next layer. If s_t =0, h_t and c_t will be initialized to 0.

4.3.1 Training and Preprocessing details

Preprocessing involves extracting visual features from videos using standard pretrained models. Not every frame is necessary since there will be very high correlation between two adjacent frames. Uniform sampling of frames was used to pick certain frames and ignore all other. If a video is divided into frames, 16 uniformly spaced frames were picked from those. Also 20 clips of 16 frames each were picked starting from the earlier 16 uniformly spaced frames.

The 16 frames are then fed to ResNet-152 for feature extraction. A vector of dimension 2048 is created by Resnet-152. Frames will encoder the appearance or static component of the frame. For encoding the motion, another network called 3D convolution network or C3D is used. Clips instead of frames are fed to C3D to obtain a feature vector of dimension 4096. Then both these feature vectors are concatenated to form a feature of dimension 6144. This vector will then be fed to the encoder for caption generation.

There is another round of feature extraction where only first and last frames are considered. Resnet-152 was used to extract appearance features from the two frames to create a feature vector of size 2048. These two frames were concatenated to form a clip of size 2 frames and fed to C3D to get a feature vector of 4096. Then these two were concatenated to form a vector of size 6144.

MSVD dataset is divided into 1200 videos for training, 100 for validation and remaining 670 videos for testing. MSR-VTT dataset is divided into 6500 videos for training, 1000 for validation and remaining 2500 videos for testing. Training is performed with minimizing cross entropy using adadelta optimizer. Learning rate is 0.0003 and used a dropout probability of 0.5 for regularization on input and hidden layer. Training was run for 100 epochs or until the improvement on validation set stops.

4.3.2 Results

MSVD and MSR-VTT datasets were used to train and test the above network. Following is the results on these datasets. Two types of data- one was representing every frame and the other was representing only first and last frame. Intuitively, there will be a drop in accuracy since network is not shown everything that is happening in the video.

Dataset	B@4	METEOR	CIDEr
MSVD	0.44	0.33	0.65
MSR-VTT	0.36	0.25	0.31

Table 4.1: Using encoded data representing every frame

Dataset	B@4	METEOR	CIDEr
MSVD	0.34	0.25	0.45
MSR-VTT	0.30	0.21	0.31

Table 4.2: Using encoded data representing first and last frame

Generated caption	Reference caption
a woman is holding a baby	a man and a woman are talking
a man is pouring water into a pot	the person is cooking
a cat is playing with a ball	a cat walks across the grass
a man is doing exercise	a man is doing bench press
a man is drinking water	a tired man is drinking juice
a man is shooting a gun	a man is shooting a gun
a baby is crying	woman is trying to calm the baby
a man is doing exercise	a man is jumping
a woman is talking on a phone	a mobile phone commercial
a frog is eating	A frog is eating a lizard

Table 4.3: Generated and reference captions. Generated captions are using features representing every frame

From the above results, it is clear that there is a significant drop in accuracy between the two type of generated captions. It will be difficult to bridge a gap that large using teacher-student network. So, we tried another network that gave better results, close to state of the art and does not have a huge gap the two types of generated captions.

4.4 Semantic Compositional Network for Visual Captioning

Proposed by Gan *et al.* (2016) Semantic Compositional Network takes visual tags along video features for caption generation. Detecting explicit semantic concepts encoded in an image, and adding this high-level semantic information into the CNN-LSTM framework, has proven to improve performance significantly.

Similar to the conventional CNN-LSTM based image captioning framework, a CNN is used to extract the visual feature vector, which is then fed into a LSTM for generating the image caption . However, unlike the conventional LSTM, the SCN extends each weight matrix of the conventional LSTM to an ensemble of tag-dependent weight matrices, subject to the probabilities that the tags are present in the image.

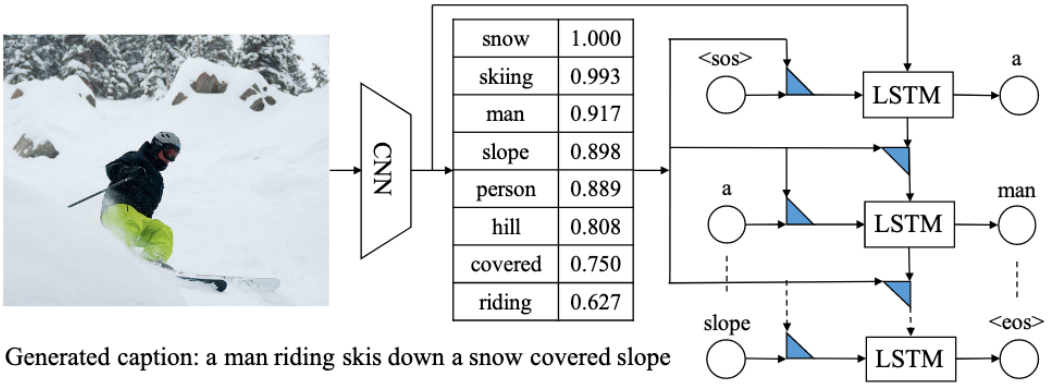

Figure 4.4: Semantic Compositional Network

4.4.1 Semantic tags generation

In order to detect a tag from an image, we first select a set of tags from the caption text in the training set. We use the K most common words in the training captions to determine the vocabulary of tags, which includes the most frequent nouns, actions or verbs.

Here K=300 is a user's choice. In order to predict semantic concepts given a test image, we treat this problem as a multi-label classification task. Suppose there are N training

examples, and $y_i = [y_{i1}, ..., y_{iK}]0, 1^K$ is the label vector of the i^{th} image, where y_{ik}= 1 if the image is annotated with tag k, and y_{ik} = 0 otherwise. Let $v_i$and $s_i$represent the image feature vector and the semantic feature vector for thei_{th}image, the cost function to be minimized is

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}(y_{ik}\log s_{ik} + (1 - y_{ik})\log(1 - s_{ik}))$$

where $s_i = \sigma(f(v_i))$ is a K-dimensional vector with $s_i = [s_{i1}, ..., s_{iK}]$, $\sigma()$is the logistic sigmoid function and f(Âů)is implemented as a multilayer perceptron. In testing, for each input image, we compute a semantic-concept vectors, formed by the probabilities of all tags,computed by the semantic-concept detection model.

Tags:
person (1), cabinet (0.931), man (0.906), shelf (0.771), table (0.707), front (0.683), holding (0.662), food (0.587)

Tags:
snow(1), outdoor (0.992), covered (0.847), nature (0.812), skiing (0.61), man (0.451), pile (0.421), building (0.369)

Figure 4.5: Generated tags with their respective probabilities

4.4.2 Incorporating tags into the network

The SCN creates each weight matrix of the conventional LSTM to be an ensemble of a set of tag-dependent weight matrices, subjective to the probabilities that the tags

are present in the image. Specifically, the SCN-LSTM computes the hidden states as follows,

$$h_t = \sigma(W(s)x_{t-1} + U(s)h_{t-1} + z)$$
$$z = I(t = 1) \cdot Cv$$

$W(s)$and $U(s)$ are ensembles of tag-dependent weight matrices, subjective to the probabilities that the tags are present in the image. Every weight matrix in a conventional LSTM will be a 2D slice of the weight matrix being trained in this network. There is a weight matrix for each tag and is a linear combination of these matrix that is evaluated for each image with coefficients being the probability of these tags present in that particular image. Following are the formulas governing tag dependent weight matrices.

$$W(s) = \sum_{k=1}^{K} s_k W_T[k]$$
$$U(s) = \sum_{k=1}^{K} s_k U_T[k]$$

Here, $s \in \mathcal{R}^K$, we define two weight tensors $W_T \in \mathcal{R}^{n_h \times n_x \times K}$ and $U_T \in \mathcal{R}^{n_h \times n_h \times K}$, where n_h is the number of hidden units and n_x is the dimension of word embedding. Observing the above tag dependent weight matrices indicate that training this network is equivalent to training K independent LSTMs.

4.4.3 Results

MSVD and MSR-VTT datasets were used to train and test the above network. Following is the results on these datasets.

Dataset	B@4	METEOR	CIDEr
MSVD	0.51	0.59	0.78
MSR-VTT	0.39	0.45	0.37

Table 4.4: Using encoded data representing every frame

Dataset	B@4	METEOR	CIDEr
MSVD	0.48	0.33	0.53
MSR-VTT	0.35	0.24	0.28

Table 4.5: Using encoded data representing first and last frame

Generated caption	Reference caption
a man and a woman are talking	a family is having conversation
a man is surfing in the water	a woman surfing in the ocean
a man is showing how to use a toy	a chef cutting bell pepper and crushing garlic
a scene from a movie is shown	a trailer for a movie is shown
a man is talking about makeup	man getting nose treatment
two men are wrestling	boys are wrestling in front of a crowd
someone is playing a game	a cartoon jumping on flowers
a man is running on a track	a woman is running in a meet
a man is being interviewed	two men are talking about something

Table 4.6: Generated and reference captions. Generated captions are using features representing every frame

CHAPTER 5

Teacher-Student Network results

In the last chapter, two networks were tested for captioning accuracy when each frame is fed and when only first and last frame is fed. Choosing the best of two networks(SCN) as a teacher, following results were generated.

5.1 Results

The loss function given in the original paper Bhardwaj and Khapra (2018) wasn't appropriate for captioning. L_2 loss with the feature representation of teacher and student network was employed. But it turned out to give a low BLEU score of 0.25. Following is the cost function plot while using $L2$ loss.

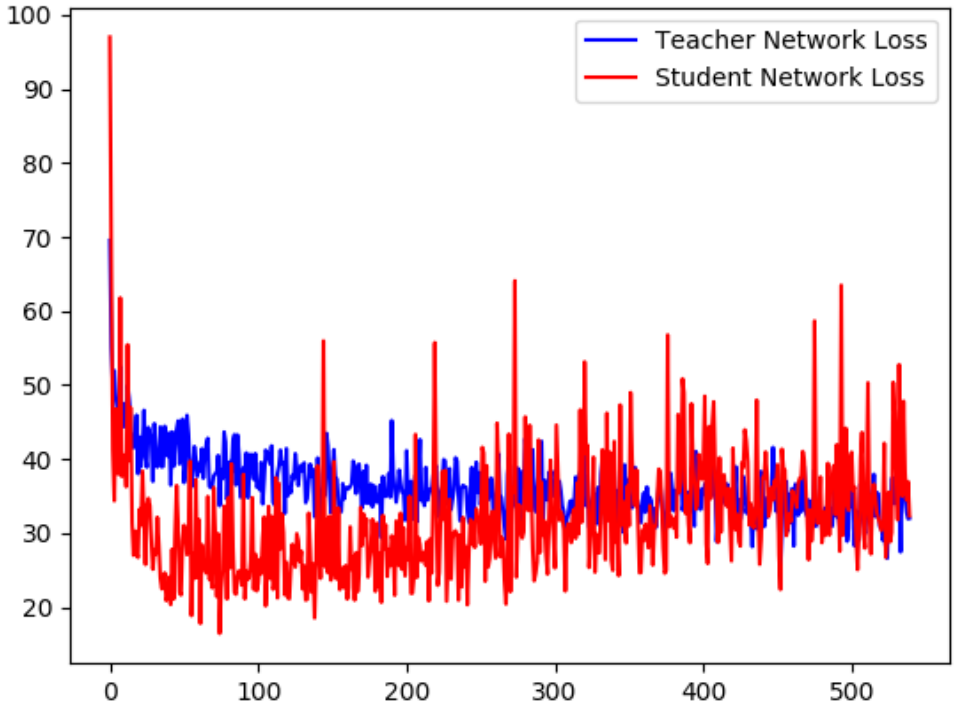

Figure 5.1: Cross entropy loss(Teacher) and L_2 loss(Student)

The next loss function employed was cross entropy loss with softmax predictions of student and teacher network. Along with that, the intermediate features encoded in LSTMs was taken with a hyperparameter λ to make the two losses comparable. The result was a BLEU score of 0.36. A slight improvement over absence of Teacher-Student network. Following is the plot of cost function. So, there was an improvement of $\approx 3\%$ in BLEU score.

Aim of this thesis was to generate captions with better BLEU-4 score using a teacher-student network. SCN was chosen as the teacher network since it had comparable to state of the art results on both MSVD and MSR-VTT datasets. Following is the training loss of teacher-student network with MSR-VTT dataset. Student loss is

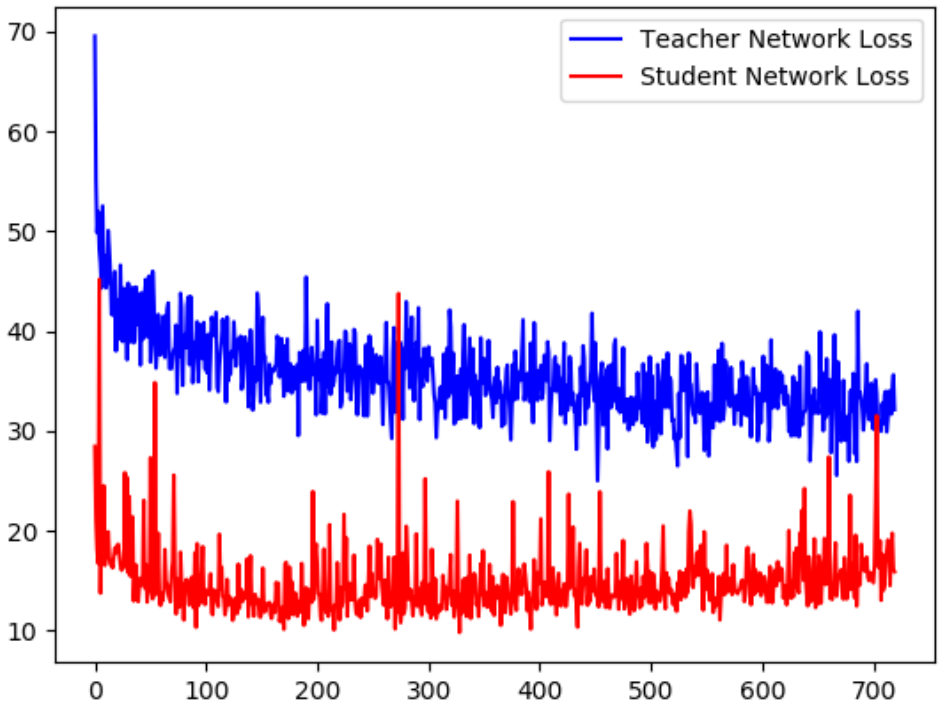

Figure 5.2: Cross entropy loss(Teacher) and cross entropy loss(student)

less than teacher loss since the loss function is different for student network. Following

Dataset	B@4	METEOR	CIDEr
MSVD	0.49	0.33	0.75
MSR-VTT	0.36	0.244	0.297

Table 5.1: Using encoded data representing first and last frame

is the comparison between reference captions and generated captions on MSVD and MSR-VTT datasets.

Reference caption: A man is cutting paper using a pair of scissors
Generated caption: A man is cutting something

Reference caption: An animal is captured in a bag
Generated caption: An animal is playing

5.2 Contribution

The core idea of Teacher-Student network was adopted for video captioning. As a result, there was a need to choose a teacher network that is able to generate quality captions even with less number of frames. Two networks were tested through captioning by feeding each frame and feeding only two frames. Semantic Compositional Network seemed to give good BLEU score even when only two frames were fed as input. So, SCN was chosen as the teacher network. There is an improvement of approximately 3% in BLEU score on the two datasets used.

5.3 Conclusion and Future directions

Video captioning using only first and last frame is advantageous mainly for mobile devices(processing power is limited) since preprocessing is reduced tremendously and

Reference caption: Someone is cutting a fruit using knife
Generated caption: A woman is chopping fruit

Reference caption: A women is cooking in a cooking show
Generated caption: A women is cooking

Reference caption: A cat and a dog are eating from the same bowl
Generated caption: A dog and a cat are playing

Figure 5.3: Generated and reference captions

also number of LSTM computations is reduced. The current Video captioning datasets are not suitable for captioning using few frames. In order to caption the "interpolated" visual features, the dataset should have a cause-effect relationships. There are basically no datasets that ensure such property in videos and hence the network cannot reach to its full potential with current datasets.

Loss function for Teacher-Student network can be improved for better performance. Trying out different loss functions might improve BLEU score. Teacher-Student network can also be used with a more compact student network. Measuring accuracy Vs complexity of neural network can give an insight regarding actual complexity required for a given task. Spacial attention mechanisms can be used to improve the results since temporal attention is not possible with only two frames.

Boundary aware network can be used to generate Video to Paragraph captioning since it summarizes one shot and reinitializes LSTM for next shot. Semantic Compositional Network can be modified with spatial attention mechanisms to further improve accuracy.

REFERENCES

1. **Bahdanau, D.**, **K. Cho**, and **Y. Bengio**, Neural machine translation by jointly learning to align and translate. *In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2015. URL http://arxiv.org/abs/1409.0473.

2. **Baraldi, L.**, **C. Grana**, and **R. Cucchiara**, Hierarchical boundary-aware neural encoder for video captioning. *In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017. ISSN 1063-6919.

3. **Bhardwaj, S.** and **M. M. Khapra** (2018). I have seen enough: A teacher student network for video classification using fewer frames. *CoRR*, **abs/1805.04668**. URL http://arxiv.org/abs/1805.04668.

4. **Deng, J.**, **W. Dong**, **R. Socher**, **L. Li**, and **and**, Imagenet: A large-scale hierarchical image database. *In 2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009. ISSN 1063-6919.

5. **Donahue, J.**, **L. A. Hendricks**, **S. Guadarrama**, **M. Rohrbach**, **S. Venugopalan**, **K. Saenko**, and **T. Darrell** (2014). Long-term recurrent convolutional networks for visual recognition and description. *CoRR*, **abs/1411.4389**. URL http://arxiv.org/abs/1411.4389.

6. **Gan, Z.**, **C. Gan**, **X. He**, **Y. Pu**, **K. Tran**, **J. Gao**, **L. Carin**, and **L. Deng** (2016). Semantic compositional networks for visual captioning. *CoRR*, **abs/1611.08002**. URL http://arxiv.org/abs/1611.08002.

7. **He, K.**, **X. Zhang**, **S. Ren**, and **J. Sun**, Deep residual learning for image recognition. *In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016. ISSN 1063-6919.

8. **Hochreiter, S.** and **J. Schmidhuber** (1997). Long short-term memory. *Neural Comput.*, **9**(8), 1735–1780. ISSN 0899-7667. URL http://dx.doi.org/10.1162/neco.1997.9.8.1735.

9. **Lavie, A.** and **A. Agarwal**, Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. *In Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07. Association for Computational Linguistics, Stroudsburg, PA, USA, 2007. URL http://dl.acm.org/citation.cfm?id=1626355.1626389.

10. **LeCun, Y.**, **P. Haffner**, **L. Bottou**, and **Y. Bengio**, Object recognition with gradient-based learning. *In Shape, Contour and Grouping in Computer Vision*. 1999. URL https://doi.org/10.1007/3-540-46805-6_19.

11. **Lin, C.-Y.**, ROUGE: A package for automatic evaluation of summaries. *In Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*. Association for Computational Linguistics, Barcelona, Spain, 2004. URL https://www.aclweb.org/anthology/W04-1013.

12. **Mikolov, T.**, **K. Chen**, **G. S. Corrado**, and **J. Dean** (2013). Efficient estimation of word representations in vector space. URL http://arxiv.org/abs/1301.3781.

13. **Pan, P.**, **Z. Xu**, **Y. Yang**, **F. Wu**, and **Y. Zhuang** (2015*a*). Hierarchical recurrent neural encoder for video representation with application to captioning. *CoRR*, **abs/1511.03476**. URL http://arxiv.org/abs/1511.03476.

14. **Pan, Y.**, **T. Mei**, **T. Yao**, **H. Li**, and **Y. Rui** (2015*b*). Jointly modeling embedding and translation to bridge video and language. *CoRR*, **abs/1505.01861**. URL http://arxiv.org/abs/1505.01861.

15. **Papineni, K.**, **S. Roukos**, **T. Ward**, and **W. Zhu**, Bleu: a method for automatic evaluation of machine translation. *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA.*. 2002. URL http://www.aclweb.org/anthology/P02-1040.pdf.

16. **Rohrbach, A.**, **M. Rohrbach**, and **B. Schiele** (2015). The long-short story of movie description. *CoRR*, **abs/1506.01698**. URL http://arxiv.org/abs/1506.01698.

17. **Rumelhart, D. E.**, **G. E. Hinton**, and **R. J. Williams** (1986). Learning representations by back-propagating errors. *Nature*, **323**, 533–. URL http://dx.doi.org/10.1038/323533a0.

18. **Venugopalan, S.**, **L. A. Hendricks**, **R. J. Mooney**, and **K. Saenko** (2016). Improving lstm-based video description with linguistic knowledge mined from text. *CoRR*, **abs/1604.01729**. URL http://arxiv.org/abs/1604.01729.

19. **Venugopalan, S.**, **M. Rohrbach**, **J. Donahue**, **R. J. Mooney**, **T. Darrell**, and **K. Saenko** (2015). Sequence to sequence - video to text. *CoRR*, **abs/1505.00487**. URL http://arxiv.org/abs/1505.00487.

20. **Venugopalan, S.**, **H. Xu**, **J. Donahue**, **M. Rohrbach**, **R. J. Mooney**, and **K. Saenko** (2014). Translating videos to natural language using deep recurrent neural networks. *CoRR*, **abs/1412.4729**. URL http://arxiv.org/abs/1412.4729.

21. **Wiseman, S.** and **A. M. Rush** (2016). Sequence-to-sequence learning as beam-search optimization. *CoRR*, **abs/1606.02960**. URL http://arxiv.org/abs/1606.02960.

22. **Xu, J.**, **T. Mei**, **T. Yao**, and **Y. Rui**, Msr-vtt: A large video description dataset for bridging video and language. *In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016. ISSN 1063-6919.

23. **Yao, L.**, **A. Torabi**, **K. Cho**, **N. Ballas**, **C. Pal**, **H. Larochelle**, and **A. Courville**, Describing videos by exploiting temporal structure. *In 2015 IEEE International Conference on Computer Vision (ICCV)*. 2015. ISSN 2380-7504.

24. **Yu, H.**, **J. Wang**, **Z. Huang**, **Y. Yang**, and **W. Xu** (2015). Video paragraph captioning using hierarchical recurrent neural networks. *CoRR*, **abs/1510.07712**. URL http://arxiv.org/abs/1510.07712.