IMAGE CAPTIONING

by

B PAVAN KALYAN 411610

J GURU PAVANI 411629

R PRANEETH 411664

Under the guidance of

Dr. K. HIMA BINDU

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY ANDHRA PRADESH

TADEPALLIGUDEM-534102, INDIA

MAY - 2020

IMAGE CAPTIONING

Thesis submitted to

National Institute of Technology Andhra Pradesh

for the award of the degree

of

Bachelor of Technology

by

B Pavan Kalyan	*411610*
J Guru Pavani	*411629*
R Praneeth	*411664*

Under the guidance of

Dr. K. HIMA BINDU

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY ANDHRA PRADESH

TADEPALLIGUDEM-534102, INDIA

MAY- 2020

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)	(Signature)	(Signature)
B Pavan Kalyan	J Guru Pavani	R Praneeth
411610	411629	411664
Date: _____	Date: _____	Date: _____

CERTIFICATE

It is certified that the work contained in the thesis titled "**IMAGE CAPTIONING**" by " B Pavan Kalyan , bearing Roll No:411610 , J Guru Pavani , bearing Roll No: 411629 and R Praneeth, bearing Roll No:411664 " has been carried out under my/our supervision and that this work has not been submitted elsewhere for a degree.

Dr. K. Hima Bindu

Dept. of Computer Science and Engineering
N.I.T. Andhra Pradesh
May, 2020

ACKNOWLEDGMENTS

First and foremost, we would like to thank our guide Dr. K Hima Bindu,Department of Computer Science and Engineering,NIT Andhra Pradesh for guiding us thoughtfully and efficiently throughout this project, giving us an opportunity to work at our own pace along our own terms, while providing us with very useful directions whenever necessary.

We would also like to thank our Head Of The Department , Dr. Karthick S, Department of Computer Science and Engineering,NIT Andhra Pradesh and all the other faculty of our department for empathizing and providing all the necessary facilities throughout the work.

We offer our sincere thanks to all other persons who knowingly or unknowingly helped us complete this project.

B Pavan Kalyan **J Guru Pavani** **R Praneeth**

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

Notation	Meaning
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-term Memory
MLP	Multi Layer Perceptron
p_π	Policy Network
$\bar{V}_\Theta$	Value Network
$f_e(.)$	Linear Mapping Function used in visual semantic embedding.
$\bar{S}$	Caption
$\bar{S}_t$	t th word of the caption
I	Image
$L_{p'}$	Loss function of Policy Network
L	Pairwise Ranking loss function.
BS	Beam Search
CNN_p, RNN_p	CNN,RNN used in policy network
CNN_v,RNN_v and MLP_v	CNN,RNN and MLP used in value network
CNN_e , RNN_e	CNN,RNN in visual semantic embedding
γ	Margin cross validator
λ	hyperparameter used during testing.

ABSTRACT

Image captioning is the process of generating syntactically and semantically correct sentence of an image. It is one of the most difficult problems in computer vision due to its complexity in understanding the visual content of the image and depicting it in a natural language sentence. Recent advances in deep learning based technologies helped to handle the difficulties present in the image captioning process. Most of the state-of-the-art approaches follow an encoder decoder mechanism which sequentially predicts the words of a sentence but we use a decision making framework in the image captioning process which uses policy and value network to collaboratively form a sentence. The policy network serves as a local guidance and the value network serves as a global guidance in forming the natural language sentence. We train both the networks using actor-critic reinforcement learning model with reward using visual semantic embedding . We experimented with our model on the MSCOCO dataset which gave a good score with respect to various metrics.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

Image Captioning is the process of generating textual description of an image. It utilizes both Computer Vision and Natural Language Processing to generate the captions. It is more challenging because it not only recognises the objects present in the image, but also describes their relationship between them in a meaningful manner.

Most state-of-the-arts [16–18] follow encoder-decoder mechanisms which have been inspired from sequence to sequence models [13] in machine translation. Encoder-decoder models generally use convolutional neural network for encoding the image and recurrent neural network for decoding the visual concept to form good semantic and syntactic sentences.The neural network backpropagates the error of the output sentence compared with the ground truth sentence which is calculated by a loss function like cross entropy/maximum likelihood. In these mechanisms during training and inference they try to get the maximum high probability word based on the current word. Moreover, encoder-decoder framework uses Maximum likelihood estimation and backpropagation during training[16]. In such cases the RNN model is trained based on the previous word in the ground truth instead of its predicted word. So the model is based only on the ground-truth not by the predicted word. This situation is called exposure bias. Inorder to solve this we use a decision making framework also termed as Reinforcement Learning which is built on the top of REINFORCE algorithm. Moreover, it directly optimizes non-differentiable test metrics[9, 15, 2] and accomplishes good results. Our Model comprises policy network and value network to collaboratively determine the next predicted word at each state. The policy network gives the probability of the next word based on the current word which serves as a local guidance. The Value network calculates the reward value based on all the possible extensions of words based on current word which serves as a global guidance. This value network adjusts the policy network inorder to get a sentence that is similar to the ground truth sentence.

To learn both the policy and value networks we use deep reinforcement learning with visual semantic embedding reward. We start by pretraing the policy network with standard supervised entropy loss,and value network by mean squared loss. Then we jointly train the model using reinforcement learning which is widely used in control and gaming theory. In gaming theory we have definite targets which can be directly optimized to reach the goal whereas characterizing a suitable optimization goal for image captioning is nontrivial. So we propose an actor-critic [14]

framework comprising policy network (Actor) and value network (critic) with reward taken from visual semantic embedding [26, 27, 28, 29]. Visual semantic embedding gives a measure of similarity between the sentence and images. It gives the correctness of the generated words and serves as a global target to optimize the image captioning through reinforcement learning.

We conduct analyses on our model to know its properties and merits and also experiments on MSCOCO dataset inorder to evaluate our model on different metrics like BLUE [9], METEOR[2], CIDEr [15]. The main aim of this project is to build a reinforcement learning model for image captioning process using a policy network and a value network. To learn both of the networks we use an actor-critic reinforcement model which uses visual semantic embedding as a reward.

CHAPTER 2

LITERATURE SURVEY

Image captioning is one of the most challenging problems in computer vision. Everyday we come across numerous images without captions. Interpreting those images for a human is an easy task but captioning all of those uncaptioned images for a machine needs an automatic image captioning model. In the last few years there are many papers which have been published on image captioning problems. In this chapter we describe some of the popular image captioning approaches. We divided the models in three categories based on the framework used to generate captions: Encoder-decoder model , Attention models and mapping the image into a multimodal space..

2.1 Encoder-Decoder model:

The neural network-based model for image captioning methods is very similar to neural machine translation [13] which uses encoder-decoder framework.The encoder servers as an feature extractor which collects the visual information present in the image in to a feature representation vector and the decoder uses the feature representation vector information to generate an output sentence using Maximum Likelihood Estimation[12].

Vinyals et al.[16] proposed a model for image captioning called Neural Image Caption generator(NIC). This method uses convolutional neural network(CNN) as an encoder which converts the image features into a single feature vector .The output of CNN is used as an input to Long short-term memory(LSTM) [6], the decoder. In the caption generating process the output of CNN is included as an initial state of an LSTM. The following words are generated based on the previous output word which is used as input to generate the next word.This process continues until we get an end token of the sentence . LRCN[24] also used the same encoder-decoder model to caption an image.

2.1.1 Encoder-CNN

As we deploy only one encoder in the encoder-decoder framework ,the performance highly relies on the feature extractor(CNN). CNN used in encoder consists of a series of convolutional layers which are building blocks of CNN and then two densely connected layers which are given to decoder to generate the sentences. AlexNet, VGGNet, ResNet, and GoogleNet (also called Inception-X Net) are some of the types of feature extractors. Out of the all listed

networks ResNet wins for being computationally efficient. A clear comparison is available at [8] ,listed in Table 1.

Architecture	Top-1 Accuracy	Top-5 Accuracy	Year
Alexnet	57.1	80.2	2012
Inception	69.8	89.3	2013
VGG	70.5	91.2	2013
Resnet-50	75.2	93	2015

Table 1: Comparisons of CNN architecture.[8]

2.1.2 Decoder-LSTM

Long short-term memory(LSTM) is an RNN architecture used in the field of deep learning. It most suits for sequential data which have long term dependencies in the sentence through a memory cell.The LSTM Model is trained to predict the next word in the sentence based on the image and all predicted words so far.

For the task of image captioning, a model is required that can predict the words of the caption in a correct sequence for the given image. This can be modeled as finding the caption that maximizes the following log probability.

$$\log_p(\bar{S}/I) = \sum_{t=0}^{N} \log_p(\bar{S}_t/I, \bar{S}_0, ..., \bar{S}_{t-1}) \qquad (1)$$

Where $\bar{S}$ is the caption, $\bar{S}_t$ is the word in the caption at location t (t^{th} word)

2.2 Attention models

The attention model was built up with a goal to reproduce regular human conduct before outlining an image , individuals will be in general focus on specific regions of that picture and afterward structure a decent clarification of the relationship of objects in those regions. A similar methodology is utilized in the attention models[17, 18, 1]. Attention and its variants are in many forms like: hard, soft, bottom-up, top-down and so on .

Attention can empower our examination and debugging of neural networks. It can give practical bits of knowledge, for example which parts of the image the system is ”looking at”. Each type of attention has its own one of a kind characteristics.

There are multiple ways to implement attention, but Xu et al.[17] divided the image into a grid of regions after the CNN feature extraction, and produce one feature vector for each. These features are used in different ways for soft and hard attention:

- In the soft attention variant, each region’s feature vector receives a weight (can be interpreted as the probability of focusing at that specific region) at each time step of the decoding RNN which emphasizes the relative importance of that location in order to generate the following word. The Maximum Likelihood Estimation(followed by a softmax), which is used to calculate these weights, is a deterministic part of the computational graph and therefore can be trained end-to-end as a part of the whole system using backpropagation as usual.

- In the hard attention only a single region is sampled from the feature vectors at each time step to generate the output word which is not a differentiable function. This prevents the network training by backpropagation because of stochasticity of sampling.The accuracy is dependent upon what number of samplings are performed and the way it is sampled.

2.2.1 Image Captioning with Semantic Attention

The conventional approaches for image captioning are either top-down i.e., moving from a gist of an image which is changed over to words, or bottom-up, which generate words describing the different parts of the image and then combining them . However, an algorithm that combines both the previously mentioned approaches for image captioning and learns to selectively attend is available in [18]

Semantic attention [18]alludes to the system of concentrating on semantically significant ideas, for example objects or activities which are necessary to building an exact picture caption. In spatial attention the emphasis is put on areas of interest; yet semantic attention relates attention for the important words utilized in the caption as it’s produced.

Contrasting this work with Xu et al. [17] , their attention algorithm figures out how to take care of the particular word ideas found inside a image rather than words characterized from explicit

spatial areas.Note that a few ideas or words may not be straightforwardly identified with a particular area,for example ”exciting” which may incorporate the whole picture. This is the situation even with ideas that are not straightforwardly found in the picture, and can be extended by ”utilizing outside image information for preparing extra visual ideas just as outer content information for learning semantics between words”.So in [18] we can explicitly can give some words during training which may not be present in the image .

In 2017, Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering from Anderson et al. [1] proposed an increasingly common strategy for attention , propelled by neuroscience which investigates the distinction between Bottom-Up and Top-Down consideration, the author presented an attention technique to proficiently meld data from the two kinds.

2.3 Mapping the image into a multimodal space

karapathy et al. [24] uses inter modal alignments between the sentence and the visual data. This is a novel model based on combination images and sentences using Convolutional Neural Networks and Bidirectional Recurrent neural networks respectively. For Every image the model gets the most compatible sentence using image-sentence score and aligns its pieces to the image. For ground the pieces it uses the center of the bounding box which is created for every object in the image.

CHAPTER 3

PROPOSED METHODOLOGY

3.1 Decision making framework

Decision making is an important problem in control theory [31] , computer gaming [32] and path navigation [33], etc. In such problems, there exist agents that interact with the environment and execute a sequence of actions from the present states in order to fulfill some predefined goals.

Decision making is commonly termed as Reinforcement learning which is an area of machine learning that focuses on how you react to the environment in order to maximize the reward. The agent (Entity) interacts with the environment,executes a series of actions to move from the current state to the next state and aims to fulfil some predefined goals.The main aim of the agent is to reach a goal with maximum reward.The block diagram of this learning is shown in Fig 1.

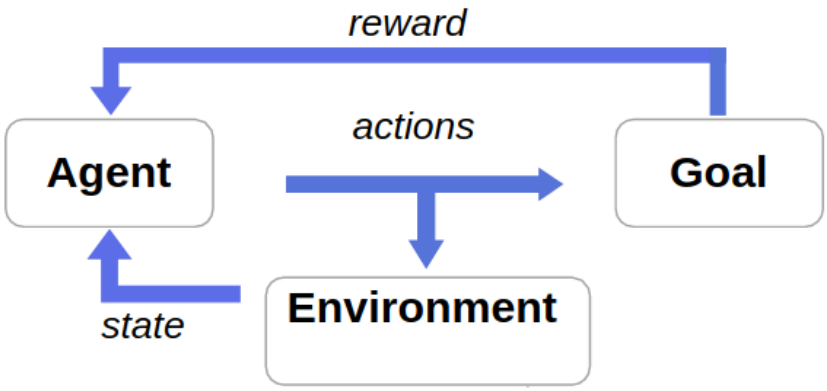


Figure 1: Block diagram of reinforcement learning [4].

3.2 Problem formulation

Reformulating the decision making framework for image captioning process

1. Goal: To generate a visual description given an image.
2. Agent: The image captioning model to learn.
3. Environment: The given image I + the words predicted so far w_1, ...w_t .
4. State: Representation of the environment at a_t .

5. Action: The word to generate at t + 1 ,$a_t = w_{t+1}$.
6. Reward: The feedback for reinforcement learning.

We propose an actor-critic model to solve image captioning problems. In the decision making framework, we have an actor-critic model comprising a policy network(actor) and a value network (critic).The policy network uses the local information for caption generation and the value network uses the global information of the image to generate a suitable caption.

3.3 Policy Network

The policy network is similar to the basic encoder decoder network. The policy network P_π gives the probability for the agent to move to the next state by performing an action at each state, $P_\pi(a_t|s_t)$ where $a_t = w_{t+1}$ and the current state $s_t = \{I, w_1, ..., w_t\}$. As shown in fig 2 it contains a CNN_p, an RNN_p network. Policy network is similar to basic encoder-decoder model [16] . The CNN_p network is used to collect visual information from the image and it is passed as an input vector to RNN_p. The RNN_p uses this information vector to generate the words in iterations to generate captions. During generation of the next appropriate word RNN_p gives the probabilities of the possible words in that iteration.These words are compared with overall probability of the sentence and choose the highest among them. To find the most suitable word we use a value network.

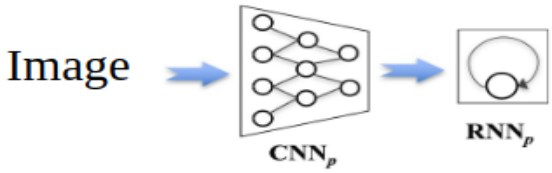


Figure 2: An illustration of policy network P_π which comprises CNN_p and RNN_p[11] .

3.4 Value Network

Value network is used to observe the global information and also for lookahead guidance(A detailed explanation provided in section 3.6). As shown in figure 3, the value network contains a CNN_v,an RNN_v and a MLP_v(Multi Layer Perceptron). In this raw image I is given as an input to CNN_v from which visual information is collected. The partially generated sentence is given as an input to RNN_v (here it acts as an encoder for sentence) to collect the semantic information from the sentence.These two informations are passed through MLP_v to get a scalar value which later used to regress the scalar reward .

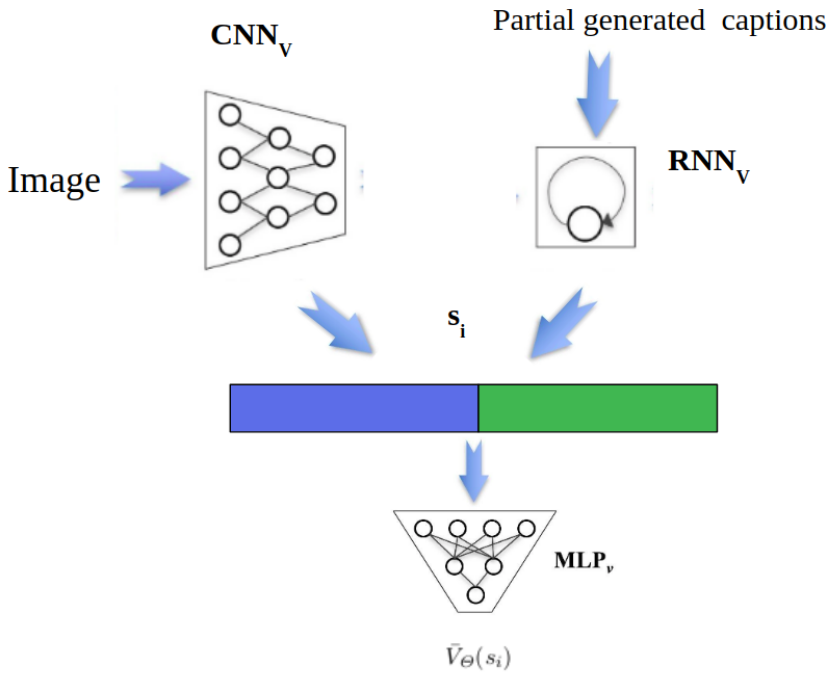


Figure 3: An illustration of Value network which comprises CNN_v and RNN_v [11].

3.5 Reward defined by visual semantic embedding

In the Reinforcement Learning framework it is more important to define a reward generation process as it leads to the optimized goal. We use visual-semantic embedding similarities as the reward.

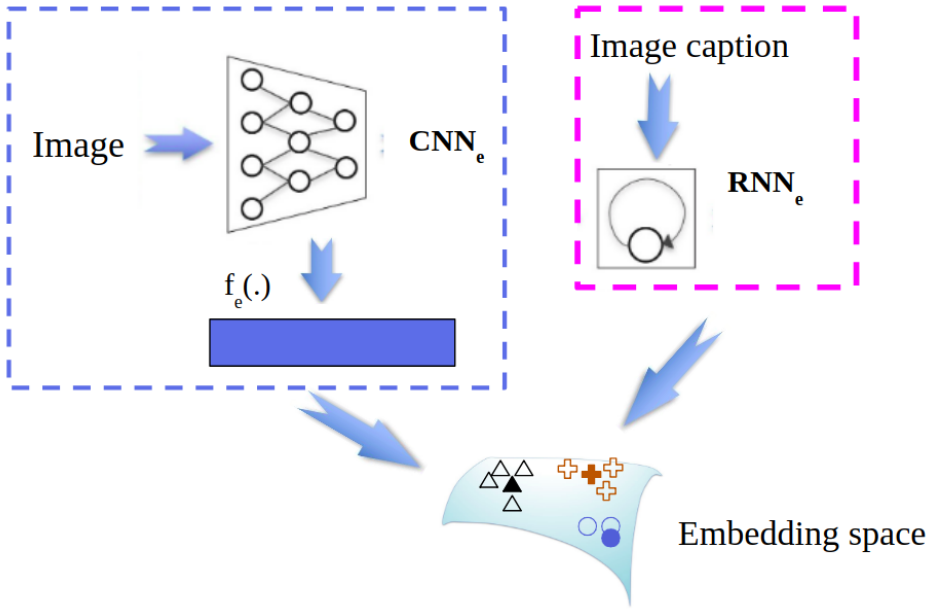


Figure 4: An illustration of visual semantic embedding consists of CNN_e , RNN_e , $f_e(.)$ [11].

Visual semantic embedding had been successfully implemented for image retrieval[28, 29, 30] and image classification [26, 27], etc.,.

The embedding model consists of a CNN, an RNN and a linear mapping layer, denoted as CNN_e , RNN_e and $f_e(.)$ .By learning the mapping of images and sentences into one semantic embedding space, it provides a measure of similarity between images and sentences. Given a sentence S, its embedding feature is represented using the last hidden state of RNN_e , h_T (S). Let v denote the feature vector of image I extracted by CNN_e, and $f_e(.)$ is the mapping function from image features to the embedding space.The embedding model is trained using the sentence-image pairs as in image captioning.The CNN_e weight is fixed and learn the RNN_e weights as well as $f_e(.)$ using pairwise ranking loss as shown in equation 3. where $\boldsymbol{\gamma}$ is margin cross-validated, for every ground truth image-sentence pair we train with negative sentence or description which is denoted as S^- for the image and vice-versa with v^-.

$$L= \Sigma_v\Sigma_{S^-} \max(0,\boldsymbol{\gamma}\text{-}f_e(v).h_T(S)+ f_e(v).h_T(S^-))+\Sigma_S\Sigma_{v^-}\max(0, \boldsymbol{\gamma}\text{-} f_e(v).h_T(S)+ f_e(v^-).h_T(S)) \quad (2)$$

Given an image with feature v_f , we define the reward of a generated sentence S to be the Cosine similarity between S and v_f

$$r = \frac{f_e(v_f).h_T(S)}{||f_e(v_f)||.||h_T(S)||} \quad (3)$$

3.6 Training Policy and Value Networks

The training involves individual training and joint training. In individual training,the policy network p_π is trained using standard supervised learning with cross entropy loss , where the loss function as defined as in equation 4 and the value network V_Θ is trained by minimizing the mean squared loss as shown in equation 5

$$L_{p'} = -\log\ p(w_1, ..., w_T|\mathbf{I};\pi) = -\sum_{t=1}^{T} \log\ p_\pi(a_t|s_t). \quad (4)$$

$$Mean\ squared\ loss = ||\bar{V}_{\Theta}(s_i) - r||^2 \tag{5}$$

where r is the final reward of the generated sentence and s_i denotes a randomly selected state in the generating process. For one generated sentence, successive states are strongly correlated, differing by just one word, but the regression target is shared for each entire captioning process. Thus, random sampling of one single state from each distinct sentence prevents overfitting.

In the next step,both p_π and $\bar{V}_\Theta$ are jointly trained using deep reinforcement learning (RL). The parameters p_π, $\bar{V}_\Theta$ are learnt by maximizing the total reward which the agent can expect when interacting with the environment.

This approach can be viewed as an actor-critic architecture where the policy network is the actor and value network is the critic. However, reinforcement learning in image captioning is difficult to train, because of the enormous action space contrasting to other decision-making problems. To handle this problem, we apply curriculum learning[3] to train the actor-critic model. In order to gradually teach the model to deliver stable sentences, training samples are provided with gradually more difficulty: iteratively fix the first $(T - j \times \Delta)$ words with cross entropy loss and let the actor-critic model train with the remaining $j \times \Delta$ words, for j = 1, 2, ..., until reinforcement learning is used to train the whole sentence where T is the length of the sentence.

3.7 Inference

Decoding the most probable output sentence for a given image includes searching through all the possible output sentences based on their likelihood.The vocabulary often includes hundreds of thousands of words which is very huge. Therefore ,the Searching problem is exponential in the length of the output sentence which is intractable(NP-complete).

In practice, Vinyals et al.[16] uses Beam search(BS). BS is a heuristic search that expands upon the greedy search and returns the list of most likely output sentences. It is the most prevalent method for decoding in existing image captioning approaches, which stores the top-K highly scoring candidates at each time step. Here K is the beam width.By searching through specific combinations of words, and creating different possible outputs, beam search constructs a whole sentence without relying too heavily on any individual word from the ones which the RNN may

generate at any specific time step. Let us denote the set of K sequences held by BS at time t as $\bar{W}_{[t]} = \{ \bar{w}_{1,[t]}, \bar{w}_{2,[t]}, \bar{w}_{3,[t]}, ..., \bar{w}_{k,[t]} \}$ where each sequence are the generated words until then, $\bar{w}_{k,[t]} = \{ \bar{w}_{k,1}, ..., \bar{w}_{k,t} \}$. At each time step t, BS considers all possible single word extensions of these beams, given by the set $\bar{W}_{k+t} = \bar{W}_{[t]} \times Y$, and selects the top K most scoring extensions as the new beam sequences $\bar{W}_{[t+1]}$ and Y is defined as an action space which is a dictionary containing all the possible words to form a sentence.

$$\bar{W}_{[t+1]} = \underset{\bar{w}_k,[t+1]}{argtopK}\, S(\bar{w}_{k,[t+1]}) \tag{6}$$

But BS uses a greedy approach so it uses only local information and is sensitive to beam sizes. In order to better utilize the global information we proposed a new lookahead inference mechanism which consolidates the local guidance of policy network and the global guidance of value network. The learned value network gives a lookahead assessment to every decision, which can supplement the policy network and collaboratively form a sentence. i.e.,

$$S(\bar{w}_{k,[t+1]}) = S(\{\bar{w}_{k,[t]}, \bar{w}_{k,t+1}) = S(\bar{w}_{k,[t]}) + \lambda log p_\pi(a_t|s_t) + (1-\lambda)\bar{v}_\Theta(\{s_t, \bar{w}_{k,t+1}\}) \tag{7}$$

where $S(\bar{w}_{k,[t+1]})$ is the score of combining the present sequence $\bar{w}_{k,[t]}$ with $\bar{w}_{k,t+1}$, $\log P_\pi(a_t|s_t)$ and $(s_t, \bar{w}_{k,t+1})$ denotes the confidence of the policy network and evaluation of the value network for the state expecting $\bar{W}_{k,t+1}$respectively. λ is a hyperparameter which combines both policy and value network.

CHAPTER 4

DATASETS AND EVALUATION METRICS

4.1 Benchmark Datasets

Three benchmark datasets are available for evaluating image captioning methods.the datasets are Flickr8K[7] , Flickr30k [19]and Microsoft COCO [5] Caption dataset.

Flickr8K **:** This dataset contains the images taken from Flickr website.It contains 8,000 images.Each image was annotated by five sentences,based on crowd sourcing service.This dataset mainly contains humans and animals.The training data includes 6,000 images,testing and validation includes 1,000 images each respectively.

Flickr30K **:** This dataset is the extension of Flickr8K dataset.It contains about 31,783 annotated images.Each image is annotated with five sentences written particularly for that image.It mainly concentrates on humans and their activities.It does not provide any particular split of training and validation. So, researchers can split the dataset into training,testing and validation sets according to their interest.

MS COCO dataset **:** Microsoft COCO dataset is formed with the images collected from complex everyday scenes with objects that are common in our life.It has 123,287 images in total, of which 82,783 and 40,504 are used for training and validation, respectively. Later it is updated and now it is divided as 118K and 5K for training and validation respectively. Each image has five human written captions.

4.2 Evaluation Metrics

Assessing machine interpretation is considerably more intricate task.Human assessment is broad however expensive.Human assessment takes a great deal of time and for this reason,three measurements have been proposed for machine interpretation.

BLEU **:** Bilingual Evaluation Understudy,in short BLEU[9].BLEU is a calculation for assessing the nature of content which has been machine-translated starting with one normal language then onto the next.The BLEU measurements scores an interpretation on a size of 0 to 1.The value indicates how similar the candidate text is to the reference texts.BLEU checks the amount of matches by differentiating n-grams of the contender and the n-grams of the reference

interpretation.Put Simply,it gauges what number of words covered in a given interpretation when contrasted with a reference interpretation.

CIDEr **:** Consensus-based Image Description Evaluation,in short CIDEr[15].This worldview involves of three principle parts:a new triplet-based technique for gathering human comments to gauge consensus,a new automated metric that catches agreement and to precisely gauge accord, we gather two new assessment datasets containing 50 depictions for each picture – PASCAL-50S and ABSTRACT-50S. The PASCAL-50S dataset is based on the well known UIUC Pascal Sentence Dataset, which has 5 portrayals for every picture. This dataset has been utilized for both training and testing in various works.The ABSTRACT-50S dataset is based on the dataset of Zitnick and Parikh..CIDEr estimates the closeness of produced sentence against a lot of ground truth sentences composed by humans,which shows high concurrence with accord as surveyed by people.

METEOR **:** METEOR[2],a programmed metric for machine interpretation assessment depends on a summed up idea of unigram coordinating between the machine delivered translation.It assesses an interpretation by computing a score dependent on express word-to-word matches between the interpretation and a reference interpretation.If more than one reference interpretation is accessible, the given interpretation is scored against each reference independently, and the best score is accounted for.METEOR is shown to have elevated levels of connection with human decisions of interpretation quality,significantly beating BLEU metric.METEOR replaces the precision and recall computation,with a weighted f-score dependent on mapping unigrams .It likewise also introduces penalty function for incorrect word order.

CHAPTER 5

EXPERIMENTAL PROCEDURE

In this section we explain about the experiments performed to evaluate our proposed methodology.

5.1 Network architecture

As shown in figure 2 and 3 our policy and value network contains both CNN and RNN architectures. First we train them independently. We used VGG-16[20] for CNN architecture and LSTM[6] for RNN architecture. Input nodes and hidden dimensions are both set to 512 dimensions. In the value network as shown in figure 3 , we used MLP_v to regress the reward, having 1024 dimensions in input layer and 512 dimensions in output layer. As shown in figure 3 we used a state a_t to concatenate both visual and semantic features. The semantic features taken from RNN_v is 512 dimensions from hidden state. The visual feature is taken from 4096 dimension CNN_v output and mapped to 512 dimension embedded feature vector. Hence the dimension of a_t is 1024 dimension.

5.2 Visual semantic embedding

Visual semantic embedding measures the similarity between the sentences and images by mapping them into a similar dimensional space. We used VGG-16[20] for CNN_e and GRU[22] for RNN_e. Image vector v in equation 2 is taken from VGG-16 consists of a 4096 dimension layer. GRU is set to 300dimension for input node and 1024 dimension for hidden state dimension. The feature mapping layer $f_e(.)$ used a 4096x1024 linear mapping layer. γ used in equation 2 is set as 0.2

5.3 Individual Training of the agent

a) We first train the policy and value network differently.
    1) As shown in figure 2 we train the policy network using cross entropy loss mentioned in equation 4.

        i) Number of epochs are 50,000 with batch size of 100.

        ii) Training data size :50,000 images.

iii) We used Adam[21] optimizer algorithm for updating the network.

iv) We stored our network parameters in a .pt file .

2) As shown in figure 4 we train the visual semantic embedding using pairwise ranking loss mentioned in equation 2.

i) Number of epochs are 50,000 with batch size of 50.

ii) Training data size :50,000 images.

iii) We used Adam [21] optimizer algorithm for updating the network.

iv) We stored our network parameters in a .pt file .

3) We get the rewards by using cosine similarity between the image and the sentence.

4) As shown in figure 3 we train the value network using the mentioned equation 5 by getting the reward r for each image and randomly selected state at the generated sentence to avoid exposure bias .

i) Number of epochs are 50,000 with batch size of 50.

ii) Training data size :50,000 images.

iii) We used Adam [21] optimizer algorithm for updating the network.

iv) We stored our network parameters in a .pt file .

b) After individual training of policy and value network and saving them in .pt files we train both the networks using actor-critic reinforcement learning model by using curriculum learning initially setting Δ to 2 and then multiply it by 2 until it reaches 16.

i) Number of epochs are 100 with episodes 50.

Implementation details

Platform : Google Colab with python 3.

Libraries and packages :

- Numpy 1.16.2
- torch 1.2.0

- nltk 3.4
- pycocoevalcap 1.0
- h5py 2.9.0
- matplotlib 3.0.3
- future 0.17.1
- imageio 2.6.1
- torch summary 1.5.1.

Execution time: 6 hours for individual training of all networks + 2 hours for joint training for all the 50,000 images

RAM : 12GB with GPU.

5.4 Testing

After training (individual + joint) we use a look ahead inference mechanism with λ=0.4 for generating a natural language sentence for a new image . We used 100 random images of the validation set of MSCOCO images to test our model with respect to various evaluation metrics like BLEU, CIDEr, METEOR and presented the results in chapter 6.

CHAPTER 6

RESULTS AND DESCRIPTION

In Table 2, We provided a summary of the results of our proposed methodology and existing methods.The values of the existing models are taken from the references. Some of the metrics are missing as the paper authors didn't provide the values for them in the papers [16,23,25]. Due to limitations in RAM we used a limited number of images (50,000 images out of 82,782 images) for training and testing.The values provided are the result of random 100 images taken from the validation set of MSCOCO . Due to the RAM limitations the scores that we got are pretty close to m-RNN[23] . In comparison to other papers except [24] our model showed significant improvements with respect to the metrics mentioned in Table 2.

Method	**BLEU-1**	**METEOR**	**CIDEr**
NIC[16]	0.666	-	-
m-RNN [23]	0.67	-	-
BRNN[24]	0.625	0.195	66.0
LRCN[25]	0.628	-	-
our model	0.672	0.23	0.82

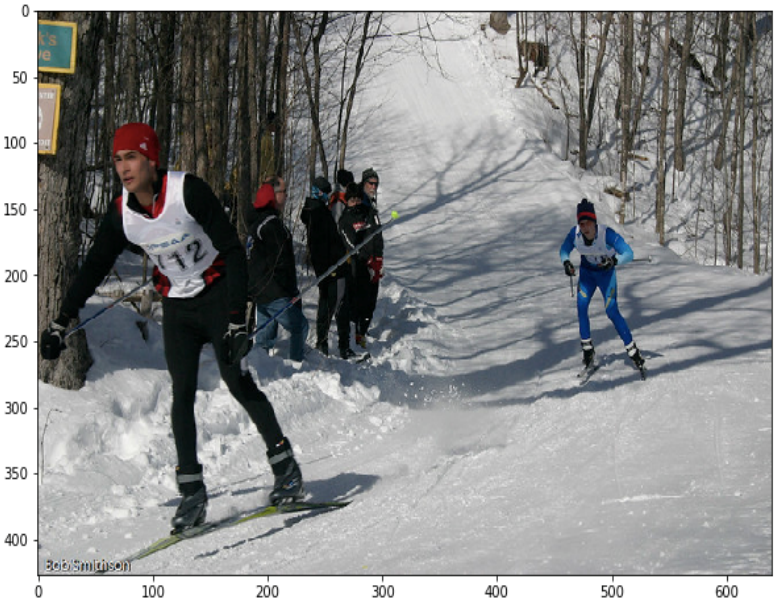
Table 2: Comparison of previous image captioning models with our model

Following are some of the captions generated using our model compared with Encoder-decoder model [16].

http://farm9.staticflickr.com/8298/7983480976_03416ab1ab_z.jpg
Reinforcement learning: <START> a group of young people walking down a street <END>
Encoder-Decoder: <START> a man is walking <UNK> with a horse <END>

http://farm1.staticflickr.com/106/279054270_14070b5c25_z.jpg
Reinforcement learning: <START> a street stop sign that is in the middle of the street <END>
Encoder-Decoder: <START> a street sign that <UNK> <UNK> not cross the road <END>

http://farm4.staticflickr.com/3613/3378688889_4e9d59ed8d_z.jpg
Reinforcement learning: <START> a man hitting a baseball during a baseball game <END>
Encoder-Decoder: <START> a man in front of a walking holding a baseball bat <END>



http://farm6.staticflickr.com/5134/5500926283_d9938bd151_z.jpg
Reinforcement learning: <START> people are posing with their skis on the snow <END>
Encoder-Decoder: <START> a couple of people on a snowy hill <END>

http://farm2.staticflickr.com/1217/896262702_334f0e78ac_z.jpg
Reinforcement learning: <START> there is a bunch of books laying on a desk <END>
Encoder-Decoder: <START> a small girl with a laptop on a tennis court <END>

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

Image captioning is an exciting activity and raises extreme challenges among scientists. There are an ever increasing number of researchers who are choosing to investigate this study field, so the amount of data is continually expanding. In this work, we presented a decision making framework for image captioning which is different from the previous encoder-decoder model , our method comprises a policy network and a value network which serves as local and global guidance respectively. To learn the two networks, we use an actor-critic model with visual semantic embedding as a reward. We performed detailed analysis on our model in understanding its metrics and properties. We understood that to output an effective sentence we should not only use local information but also should take account of global information provided by the model.

Our future work involves improving architectures of our networks and also improving embedding measures in our reward system.

REFERENCES

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and VQA. CoRR, abs/1707.07998, 2017.

[2] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. 01 2005.

[3] Y. Bengio, J´erˆome Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. volume 60, page 6, 01 2009.

[4] Shwetha bhatt. Things to know about reinforcement learning. [Available online https://www.kdnuggets.com/2018/03/5-things-reinforcement-learning.html; accessed 02-feb-2020].

[5] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Doll´ar, and C. Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. CoRR, abs/1504.00325, 2015.

[6] Sepp Hochreiter and J¨urgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

[7] Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics (extended abstract). In IJCAI, 2013.

[8] Koustubh. ResNet, AlexNet, VGGNet, Inception: Understanding various architectures of convolutional networks. [Available online https://cvtricks.com/cnn/understand-resnet-alexnet-vgginception/; accessed 02-feb-2020].

[9] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[10] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. 2016.

[11] Zhou Ren, Xiaoyu Wang, Ning Zhang, Xutao Lv, and Li-Jia Li. Deep reinforcement learning-based image captioning with embedding reward. CoRR, abs/1704.03899, 2017.

[12] Ruslan Salakhutdinov and Hugo Larochelle. Efficient learning of deep boltzmann machines. 9:693–700, 13–15 May 2010.

[13] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. CoRR, abs/1409.3215, 2014.

[14] R. S. Sutton, D. Mcallester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In Advances in Neural Information Processing Systems 12, volume 12, pages 1057–1063. MIT Press, 2000.

[15] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensusbased image description evaluation. In CVPR, pages 4566–4575. IEEE Computer Society, 2015.

[16] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator, 2014. cite arxiv:1411.4555.

[17] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. CoRR, abs/1502.03044, 2015.

[18] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. CoRR, abs/1603.03925, 2016. 12 Pavan et al.

[19] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. Transactions of the Association for Computational Linguistics, 2:67–78, 2014.

[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization,2014.

[22]Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014

[23] J. Mao, W. Xu, Y. Yang, J. Wang, and A. Yuille. Deep captioning with multimodal recurrent neural networks. In ICLR, 2015.

[24] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions, 2014.

[25] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description, 2014

[26] A. Frome, G. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In NIPS, 2013.

[27] Z. Ren, H. Jin, Z. Lin, C. Fang, and A. Yuille. Joint image-text representation by gaussian visual-semantic embedding. In ACM Multimedia, 2016.

[28] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual semantic embeddings with multimodal neural language models. In TACL, 2015.

[29] Z. Ren, H. Jin, Z. Lin, C. Fang, and A. Yuille. Multi-instance visual semantic embedding. In arXiv:1512.06963, 2015.

[30] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui. Jointly modeling embedding and translation to bridge video and language. In CVPR, 2016.

[31] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. Nature, 518:529–533, 2015.

[32] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of go with deep neural networks and tree search. Nature, 529:484–489, 2016.

[33] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, and A. Gupta. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In arXiv:1609.05143, 2016.

APPENDIX

Feature extraction from images through CNN

Directly we downloaded the images features vectors from VGG which are stored in .h5 files. Following are commands used in colab to get the files for training and testing. Using command 1 downloaded the zip file , extracted the files using command 2and finally removed the zip file using command 3.

1) wget "http://cs231n.stanford.edu/coco_captioning.zip"
2) unzip coco_captioning.zip
3) rm coco_captioning.zip

Installing COCO API

We used the MSCOCO dataset to train our model. To support pycocoeval which is used for evaluating captions we need to install coco api . Following are the commands used to install api in colab.

!git clone https://github.com/waleedka/coco

!pip install -U setuptools

!pip install -U wheel

!make install -C coco/PythonAPI.

We used torch libraries to save our model in .pt files

Evaluation of MSCOCO captions during testing

We used open source code for evaluating the generated caption with respect to various metrics like BLUE, METEOR, CIDEr during inference . Following command is to install them in colab.

pip install git+https://github.com/salaniz/pycocoevalcap