NLP FINAL REPORT

Project: Image Caption Generation Using Deep Learning Techniques.

Before I define the problem statement, I would like to ask you to Interpret something from the given Image.

Image 1: Example of Image Captioning

What does the Image mean to you?

Maybe **A family is having dinner together and they are smiling looking at their son.**

Or **Family is having dinner together with lots of smiles on their faces.**

Sometimes it is hard to Interpret, so here comes the problem statement.

Problem statement:

Interpretation differs from person to person because everyone has their own intuition levels, and they always vary.

For example, If I show a picture to a child and ask, she will give an answer and If I show the same picture to a teenager and ask, he will give a different answer and the same as with the old man/woman. So, in all the above scenarios it is the same picture, but the thoughts and perceptions are different.

Image Captioning plays a very vital role in most of the applications these days. It has made lives easier for people. It has wider applications, and many companies are investing their time to improve it.

Before going to further details few more details.

What is Image Captioning?

It is the process of generating a textual description from an image – based on the objects and actions in the image

What is the use?

- It helps in recognizing the context of an image and annotating it with a relevant caption using deep learning and computer vision.

- It Includes the labelling of an image with English words with the help of datasets provided during model training

Applications:

Let's list a few of the applications of Image captioning. They are

1) Google Photos

Image Captioning is used to classify your photo based on the person, mountains, sea, sky etc.,

2) Facebook

Using AI to classify, segment and find patterns in pictures

3) Predicting crop yield

In the US, to predict the crop yielded in one year. The Images of agricultural crops are being captured through satellite.

4) Picasa

Using facial recognition to identify your friends and you in a group picture

5) Add Generation

Generation of ads based on age, and gender. This is possible through Image Captioning.

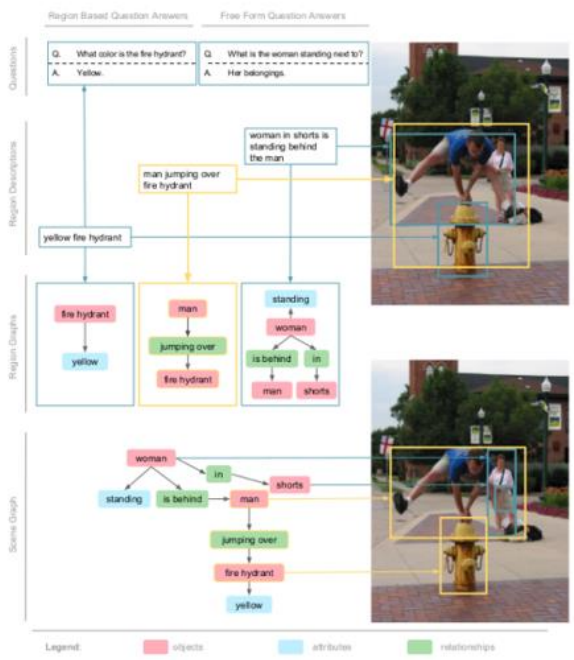
Why should we care about this solution?

This solution is making our lives easier. A lot of the latest technologies or Applications are now being dependent on Image Captioning.

There are several types of Image Captioning.

- Generating an image caption using a dataset
    1) Single sentence generation
    2) Multiple sentence / Paragraph generation
- Caption generation for a video
- Single sentence generation is the generation of a caption in a single sentence. Whereas Multiple sentence generation / Paragraph generation is generating caption in the paragraph.
- But currently, our project is based on Single sentence generation.

Let's Look into Multiple or paragraph generation and how it works



Here the Image recognizes the Relationship, Attributes and Objects. Using the above picture, we can understand how the Caption is getting generated. When you combine all the roots of the tree. That would generate the caption

The caption is: **A man is jumping over the fire hydrant which is yellow in colour and a woman in shorts is standing behind the man.**

Also, there are different kinds of Image descriptions. They are

1. Conceptual

    1) Specific: Identifying people and locations --- We are focusing on this.

    2) Generic: Related to scene and understanding

2. non-visual

3. Perceptual: From a photographer or person's point of view.

Data:

- Data for my model is Image. It should satisfy certain constraints like
- The image should be in JPG format
- The Image should be clear
- We can upload one image at a time, but you can upload multiple images in the same program.

To develop this project, we are using the Flickr dataset.

Flickr:

- Flickr has a very huge dataset for Image captioning
- 8k images in Flickr8k,2 >30k images in Flickr30k,3 with 5 descriptions per image.
- More image sentence pairs to train and test models.
- 21% of images (40% images in UIUC Pascal Sentence dataset) have static verbs like sit, stand, wear, look or no verbs.

Methods:

The methods that are being used for the Image captioning generation are

1) **CNN**
2) **RNN**

1) CNN:

It is defined as Convolutional Neural Network which is a deep learning algorithm that can take an input image, assign importance (learnable weights and biases) to various objects in the images and be able to differentiate one from the other.

- The Pre-processing required in CNN is much lower than in other classification algorithms. It extracts the features and nuances out of our image
- ImageNet dataset is used to train the CNN model called Xception. Xception is responsible for image feature extraction. These extracted features will be fed to the LSTM model which in turn generates the image caption

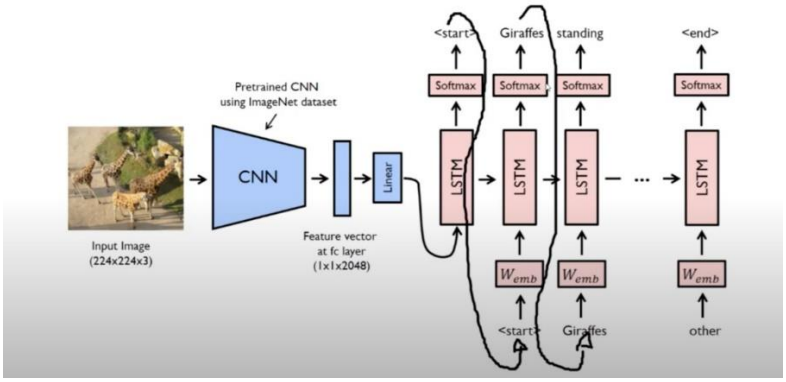


Image 1.1: How the CNN algorithm works

Reference Image: https://stats.stackexchange.com/questions/387596/image-caption-generator

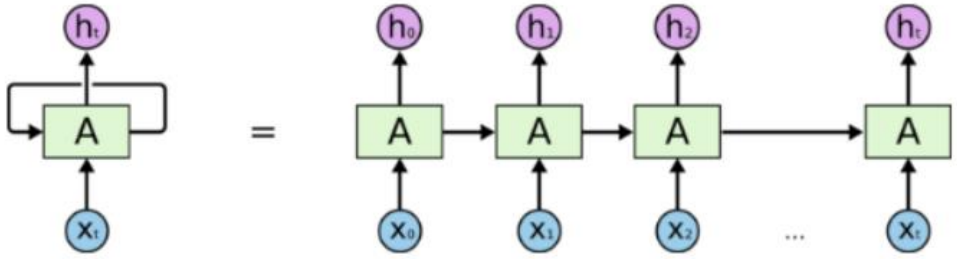
In the above model, the first LSTM cell of the decoder takes the entire image as an input. Then it reads the objects in the image like in the above example firstly we got 'giraffe', then we have 'standing', maybe next it would be 'on', then it would be 'the' then later it might generate on the 'road'. So, in the whole process from start to end we would get the whole caption that is generated for the image.

Long Short-Term Memory (LSTM) networks are a modified version of recurrent neural networks, which makes it easier to remember past data in memory**.**

In one statement: CNN is the method used to Process the Image.

2) RNN:

IT is defined as the Recurrent Neural Network. it is a generalization of a feedforward neural network that has internal memory. It is recurrent in nature as it performs the same function for every input of data while the output of the current input depends on the past computation.



- ✓ RNN can use their internal state to process sequences of inputs.
- ✓ In RNN, all the inputs are related to each other.
- ✓ It translates the features and objects given by our image-based model into natural sentences

Advantages of Recurrent Neural Network

1. It can model the sequence of data so that each sample can be assumed to be dependent on previous ones
2. Recurrent neural networks are even used with convolutional layers to extend the effective pixel neighbourhood.

I am using this RNN to create an Image Caption.

For training LSTM, we use

- ➢ For training the LSTM model, we predefine our label and target text.
- ➢ For example:

If the caption is “A man and a girl sit on the ground and eat.”

Label –

[<start>, A, man, and, a, girl, sit, on, the, ground, and, eat.]
Target – [A, man, and, a, girl, sit, on, the, ground, and, eat, ., <end>]

Experiment:

It Involves

- Get loader Program
- Modelling
- Training
- Utilities

1)Get loader Program

- This part involves the data set part. So, it decides by giving either true or false after uploading the data.
- If the Image contains in the proper format, then it would display True, else it would display false.

2)Model

- This is where we use some of the existing datasets to generate the caption.
- So, I am using the Flicker dataset to generate captions for the image uploaded. Flicker has become a standard benchmark for the sentence-based image description

Training Model

In [30]:
```
def create_weights(shape, suffix):
    return tf.Variable(tf.truncated_normal(shape, stddev=0.7), name='W_' + suffix)

def create_biases(size, suffix):
    return tf.Variable(tf.zeros([size]), name='b_' + suffix)
```

In [31]:
```
def conv_layer(inp, kernel_shape, num_channels, num_kernels, suffix):
    filter_shape = [kernel_shape[0], kernel_shape[1], num_channels, num_kernels]
    weights = create_weights(shape=filter_shape, suffix=suffix)
    biases = create_biases(num_kernels, suffix=suffix)
    layer = tf.nn.conv2d(input=inp, filter=weights, padding='SAME', strides=[1, 1, 1, 1], name='conv_' + suffix)
    layer += biases
    layer = tf.nn.relu6(layer, name='relu_' + suffix)
    #layer = tf.nn.max_pool(layer, ksize=[1, 2, 2, 1], strides=[1, 2, 2,1], padding= 'SAME')
    return layer
```

In [32]:
```
def flatten_layer(layer, suffix):
    layer_shape = layer.get_shape()
    num_features = layer_shape[1:4].num_elements()
    layer = tf.reshape(layer, [-1, num_features], name='flat_' + suffix )
    return layer
```

In [33]:
```
def dense_layer(inp, num_inputs, num_outputs, suffix, use_relu=True):
    weights = create_weights([num_inputs, num_outputs], suffix)
    biases = create_biases(num_outputs, suffix)
    layer = tf.matmul(inp, weights) + biases
    layer = tf.nn.relu(layer)
    return layer
```

3) Train Program:

- Here I am trying to train the model and generate the caption for the image I have uploaded from my local system by using the path.
- For this, I will try to generate a dataset for the image and create a caption for it. The caption of the image is generated using optimization.

4) Utilities Program

- The caption which is generated would send the result to the console and we can see the caption in it.

Results:

- The below code would train the model and my model is running successfully
- It would predict the caption for any image we feed with the help of trained captions

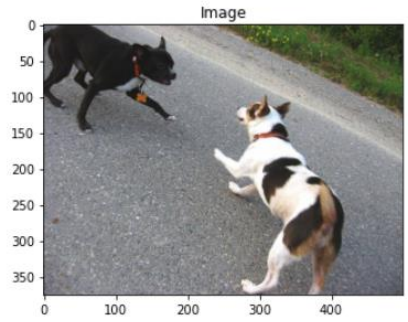
```
with tf.Session() as sess:
    sess.run(init)
    print(1)
    m = len(train_caption)
    for epoch in range(training_iters):
        total_cost = 0
        total_acc = 0
        print(2)
        for i in range(m):
            _, cst, acc = sess.run([optimizer, cost, accuracy], feed_dict = {x_caption:train_caption[i][:-1].A, x_inp:train[i
            total_cost += cst
            total_acc += acc
            print(3)
        if (epoch + 1) % display_step == 0:
            print('After ', (epoch + 1), 'iterations: Cost = ', total_cost / m, 'and Accuracy: ', total_acc * 100/ m , '%' )
    print('Optimization finished!')
    print("Let's check")
    for tests in range(num_tests):
        image_num = random.randint(0, sample_size - 1)
        caption = sess.run(final_prediction, feed_dict = {x_inp:train[image_num:image_num + 1]})
        print(caption.shape)
        caption = np.argmax(caption[:-1], 1)
        capt = ''
        print(4)
        for i in caption:
            capt += rev_dict[i] + ' '
        print('Predicted Caption:->', capt)
        orig_cap = np.argmax(train_caption[image_num:image_num + 1][0][1:-1].A, 1)
        orignalcaption = ''
        for i in orig_cap:
            orignalcaption += rev_dict[i] + ' '
        print('Orignal Caption:->', orignalcaption)
        plt.imshow(real_images[image_num])
        plt.title('Image')
        plt.show()
```

Now let me provide the images of the output screen.

It is done for training_iters = 40, it gives better results when training_iters are more.

```
Optimization finished!
Let's check
(50, 154)
4
Predicted Caption:-> A big collage glass in spotted near from front lays one by white rope water yellow red orange standing
collage uses black head entry paint pictures nearby snow through pictures blue is woman skier hands grassy get her sprinkler
lawn covered skyscraper rainbow white covered dogs building covered tent
Orignal Caption:-> A dog shakes its head near the shore , a red ball next to it .
```

```
(50, 154)
4
Predicted Caption:-> A an his waters different are Two dogs play man pink of <s> brown city her . crouches . white water bow
l boy building grass crampons the in toy playing crampons set leaps fence up chases Two crampons an baby snow brown surf sta
irs jumping grassy grass an baby
Orignal Caption:-> A black dog and a spotted dog are fighting
```



Summary:

- Image Captioning is a multimodal task which constitutes deciphering the image and describing it in natural sentences.
- I am using more than 3k images to train the model
- The data set is collected from Flickr.

Future Scope:

- I can work on Multiple sentence / Paragraph generation
- Caption generation for a video

References:

https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00571-w

https://arxiv.org/pdf/2107.06912.pdf

https://proceedings.neurips.cc/paper/2019/file/680390c55bbd9ce416d1d69a9ab4760d-Paper.pdf

https://www.hindawi.com/journals/cin/2020/3062706/

https://ieeexplore.ieee.org/document/9400209

https://paperswithcode.com/task/image-captioning

https://www.ripublication.com/ijaer18/ijaerv13n9_102.pdf