Image captioning using memory constrained LSTM with Attention

Rachit Yagnik
MT2020078
rachit.yagnik@iiitb.org

Shourabh Payal
MT2020054
shourabh.payal@iiitb.org

Sujit Kumar
MT2020106
sujit.kumar@iiitb.org

Anshul Kumar Garg
MT2020154
anshulkumar.garg@iiitb.org

Group Code: SARS

Abstract—**We will attempt to build an Image captioning system using memory constrained LSTM with attention. Image captioning is typically formulated using a generative model, creating descriptions in textual space given the input image via CNN-to-RNN.**

Index Terms—**Convolutional neural network, LSTM, Attention, Flickr**

I. Dataset

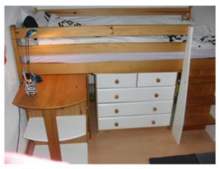
We used Flickr dataset for both training & validation and testing that 8000 Natural images that are each paired with five different captions which provide clear descriptions of the salient entities and events. Images were chosen from six different Flickr groups and tend not to contain any well known people or locations but were manually selected to depict a variety of scenes and situations. This dataset contains 8000 images each with 5 captions.

These images are distributed as follows:-

- Training Set — 6000 images
- Val Set — 1000 images
- Test Set — 1000 images

The man at bat readies to swing at the pitch while the umpire looks on.

A large bus sitting next to a very tall building.


Bunk bed with a narrow shelf sitting underneath it.

A horse carrying a large load of hay and two people sitting on it.

II. Data Pre-processing

- Random resize crop : Random resize crop is a data augmentation technique. This is done to create new training example from our existing dataset that will help our model to generalize better. This will randomly extract patches of given size. We used patch size of (224x224). So it will extract patches of size (299x299) from images randomly. It might be from top corner, bottom or centre.

- Random horizontal flip : Once we have images of size (299x299), we can choose to flip it. This is another part of data augmentation.

- Normalization : This is just input data scaling. We will normalize our tensor image with given mean and standard deviation. Both mean and standard deviation should be sequence.

III. Model

We will be dividing the problem into below modules.

- **Encoder using resnet18:** We will encode the images into 512x14x14 tensor. We chose this model because it has small size (44 Mb) and approximately 11 million parameter. This leaves room for lstm to have 14 million parameters. We are constrained by the size of the model i.e it should be below 100 Mb in size which amounts to 25 million parameters assuming float data type.

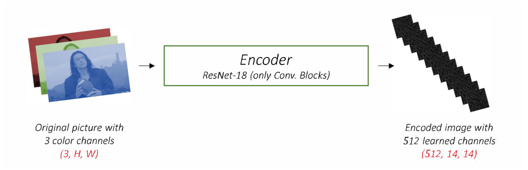
- **Attention mechanism:** Attention is way for a model to choose only those parts of the encoding that it thinks is relevant to the task at hand. we use activation function as ReLU and softmax function with dimension 1.

- **Decoder using LSTM cell:** We will use a single LSTMCell instead of the LSTM from pytorch to have more control on the behaviour modelling.

- **Teacher forcer mechanism:** Teacher forcing is a method for quickly and efficiently training recurrent neural network models that use the ground truth from a prior time step.

A. Encoder using resnet18

The Encoder encodes the input image with 3 color channels into a smaller image with "learned" channels. This smaller encoded image is a summary representation of all that's useful in the original image. Now for encoding images we use 18 layered Residual network trained on ImageNet classification task. These models progressively create smaller and smaller representations of the original image, and each subsequent representation is more "learned", with a greater number of channels. The final encoding produced by our ResNet-18 encoder has a size of 14x14 with 512 channels, i.e., a 512, 14, 14 size tensor.

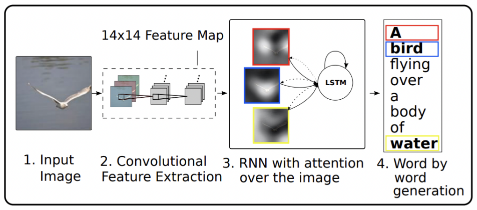


B. Attention mechanism

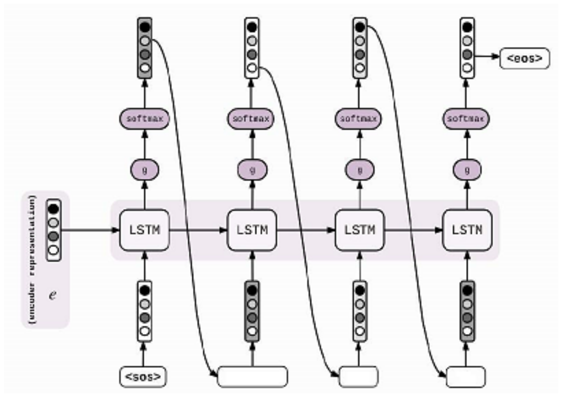
Intuitively, how would you estimate the importance of a certain part of an image? You would need to be aware of the sequence you have generated so far, so you can look at the image and decide what needs describing next.
For example, after you mention a man, it is logical to declare that he is holding a football. This is exactly what the Attention mechanism does – it considers the sequence generated thus far, and attends to the part of the image that needs describing next.

We will use soft Attention, where the weights of the pixels add up to 1. If there are P pixels in our encoded image, then at each timestep t.

$$\sum_{p}^{P} \alpha_{p,t} = 1$$


14x14 Feature Map
LSTM
A
bird
flying
over
a
body
of
water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation

C. Decoder using LSTM cell:



D. Teacher forcer mechanism:

Teacher forcing is a strategy for training recurrent neural networks that uses ground truth as input, instead of model output from a prior time step as an input.

Models that have recurrent connections from their outputs leading back into the model may be trained with teacher forcing.

Teacher forcing works by using the actual or expected output from the training dataset at the current time step y(t) as input in the next time step X(t+1), rather than the output generated by the network.

Algorithm:-

- First, we have to add a token to signal the start of the input sequence and another to signal the end of input sequence so that Algorithm can identify start and end of the input sequence.
- these signal call it as [START] and [END].
- now, we feed the model from [START] and let the model generate next word.
- if model generated output is different than actual output then we discard this generated output after calculating error. and feed next word as part of the input on the subsequent time step.
- now repeat this process for each input-output pair of words.

Teacher forcing is a fast and effective way to train a recurrent neural network that uses output from prior time steps as input to the model.

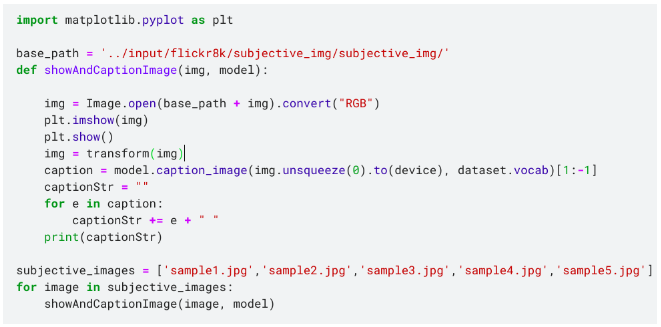
In our model, we are stopping teacher forcing after 30-40 epochs because after 30-40 epochs we are adding predicted word of model into next input so generalization will be good.

IV. Results

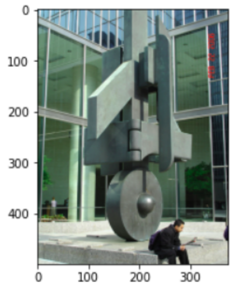
1) **Without Attention**
   Bleu score obtained was 0.111

2) **With Attention**
   We were able to achieve a **bleu score of 0.211**

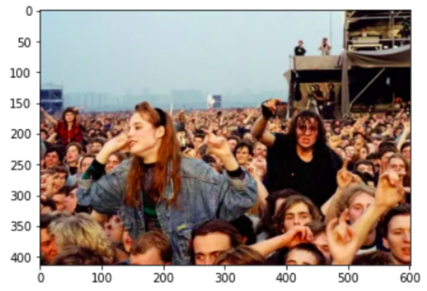
3) **Results on subjective images**
   Below are results obtained on some subjective images.

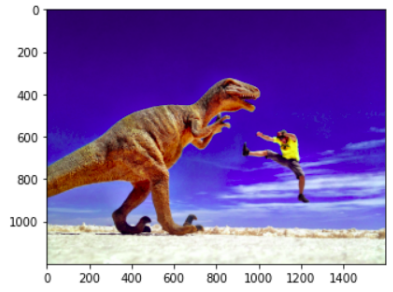


```python
import matplotlib.pyplot as plt

base_path = '../input/flickr8k/subjective_img/subjective_img/'
def showAndCaptionImage(img, model):

    img = Image.open(base_path + img).convert("RGB")
    plt.imshow(img)
    plt.show()
    img = transform(img)
    caption = model.caption_image(img.unsqueeze(0).to(device), dataset.vocab)[1:-1]
    captionStr = ""
    for e in caption:
        captionStr += e + " "
    print(captionStr)

subjective_images = ['sample1.jpg','sample2.jpg','sample3.jpg','sample4.jpg','sample5.jpg']
for image in subjective_images:
    showAndCaptionImage(image, model)
```


a man is doing a trick on a skateboard in front of a building .

a group of people are standing in front of a building .

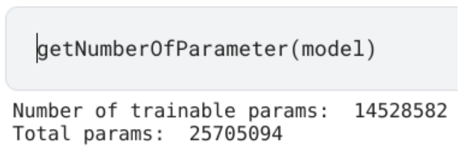

a group of people are standing in front of a large building .

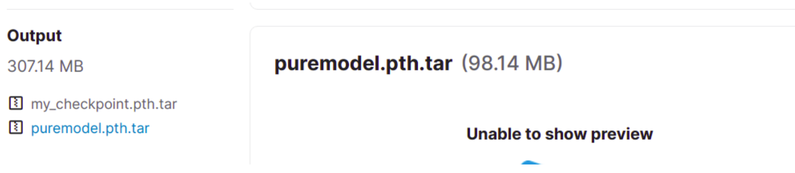

a dog is jumping into the air to catch a frisbee .

a man is standing in front of a white building with a white dog .

V. Conclusion

We were able to come up with a model under the given memory constraints and achieve a **BLEU score of 0.211** using attention mechanism.


```
getNumberOfParameter(model)

Number of trainable params:  14528582
Total params:  25705094
```


Output
307.14 MB
my_checkpoint.pth.tar
puremodel.pth.tar

puremodel.pth.tar (98.14 MB)

Unable to show preview

Challenges and Future Work

Integrate the model with an end to end system and test for practical usage. We will also try to implement the concept of beam search while captioning the image to achieve better bleu scores. We will look forward to implement transformer mechanism.

Acknowledgment

We would like to thank Professor Vishwanth G and also the Teaching Assistants for giving us the opportunity to work on this project and help us whenever we were struck by giving ideas and pointing out to specific resources that we could learn from. Also, we would like to thank the all teams for sharing their ideas and having open ended discussions. We really had a great learning experience while working on this project.

References

[1] https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html
[2] https://github.com/sgrvinod/a-PyTorch-Tutorial-to-Image-Captioning
[3] https://www.youtube.com/watch?v=y2BaTt1fxJU

Code link

Full Code link