Image Caption Generation using Attention

Gourang Patel
Masters in Data Science
Northeastern University
Boston, MA 02115
patel.gou@northeastern.edu

Sanjan Vijayakumar
Masters in Data Science
Northeastern University
Boston, MA 02115
vijayakumar.sa@northeastern.edu

Sagar Singh
Masters in Data Science
Northeastern University
Boston, MA 02115
singh.sag@northeastern.edu

Preet Pinakinbhai Shah
Masters in Computer Science
Northeastern University
Boston, MA 02115
shah.preet@northeastern.edu

1 Introduction

1.1 Problem Statement

Automatically describing the content of an image is a fundamental problem in artificial intelligence that connects computer vision and natural language processing. Some of its application includes helping visually impaired people better understand the content of images on the web.

One of the key challenge involves generating description that must capture not only the objects contained in an image, but also express how these objects relate to each other as well as their attributes and the activities they are involved in. Moreover, the above semantic knowledge has to be expressed in a natural language like English, which means that a language model is needed in addition to visual understanding.

1.2 Classic Image Captioning Method

Caption Generation is a challenging problem in Artificial Intelligence, we aim to generate a textual description for a given image. We aim to use a combined method from Computer Vision to understand the content of an image and a Language model from Natural Language Processing to transform our understanding from the text to words in a semantically oriented order.
A **"classic"** method involves encoding and extracting the features using a pretrained [VGG-16] model. Then, the decoder will be used to decode the hidden state produced by the pretrained model and produce the captions by using a Decoder(LSTM or GRU).

1.3 Problems with 'Classic' Image Captioning Model

The problem with the classical image captioning approach is when the decoder tries to predict the next word, it just captures the information of a particular portion of the image and caption it. Thus, the caption generated is not semantically correct. In general, the classic method doesn't capture the essence of the entire image. In this approach, we are using the entire hidden state h and trying to condition that over the decoder inputs to generate the words, and hence it is not able to produce different words for different parts of the image. Thus, with our implementation we are trying to resolve this issue and use "Attention"[1] mechanism which will help in capturing the semantic information from the image and capture it's real essence.

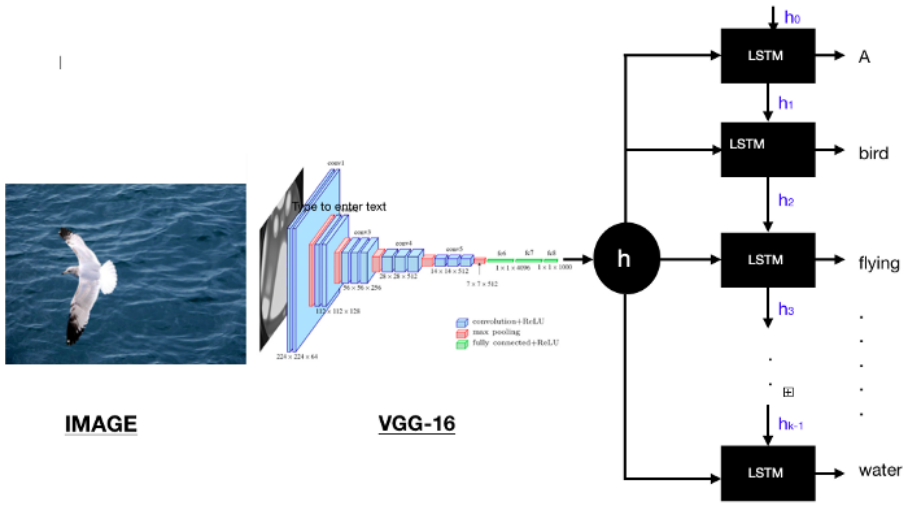

Figure 1: A classic Image Captioning Model [2]

2 Proposed Method

We are trying to encounter the problem faced by the "classic" image captioning method using the Attention mechanism in the decoder [3]. The attention mechanism will help the decoder to focus on relevant parts of the image. The decoder will only use specific parts of the image rather than conditioning on the entire hidden state h produced from the convolutional neural network.

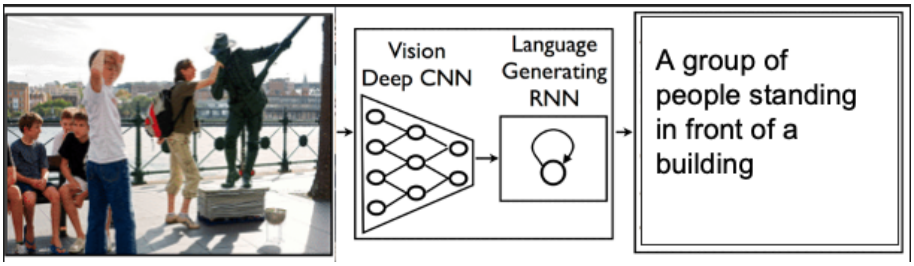

Figure 2: Architecture Overview

We can observe in Figure 3, there is one additional layer from the classic architecture, and this new layer makes the model as attention model. While predicting the next word while generating captions for an image, if we have previously predicted i words, the hidden state will be h_i. Then the model will select the relevant part of the image using the attention mechanism which will be z_i (which captures only relevant information from the image) and this will be go as an input to the LSTM. The LSTM then generates the next word and also passes on the information to the new hidden state h_{i+1}

The model architecture is inspired by the `Show, Attend and Tell` paper [1], where an attention based model was introduced to automatically learn and describe the content of images.

Given an input image we use a CNN as an image “encoder”, by first pre-training it for an image classification task and then using the last hidden layer as an input to the RNN decoder that generates complete sentences.

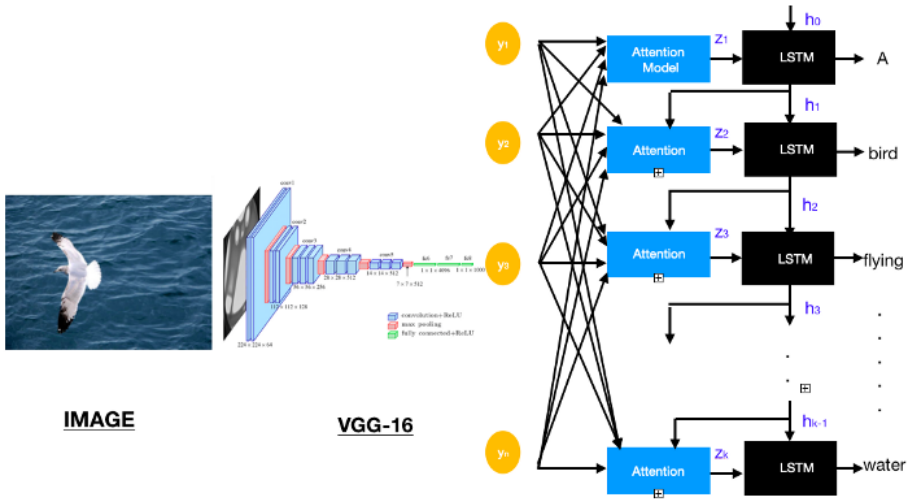

Figure 3: Image Captioning using Attention Mechanism

3 Data Preparation

3.1 Data Source

We leveraged Flickr 8K dataset consisting 5 captions for each image to train our model. The dataset contains a total of 8092 images each with 5 captions, in total we have 40460 properly labelled captions.

the white and brown dog is running over the surface of the snow
a white and brown dog is running through a snow covered field .
a dog running through snow .
a dog is running in the snow
a brown and white dog is running through the snow .

Figure 4: Data Overview

3.2 Data Preprocessing

We conducted following data Preprocessing steps

- Cleaned the captions by removing punctuations, single characters, and numeric values.
- Added start and end tags for every caption, so that model understands the start and end of each caption.
- Resized images to 224 X 224 followed by pixel normalization to suit our VGG16 image feature extraction model.
- Tokenized the captions (for example, by splitting on spaces) to obtain a vocabulary of unique words in the data.
- Padded all the sequence to be the same length as the longest one.

4 Model/Methodology

4.1 Pre-trained Image Model(VGG-16)

We have used VGG-16 architecture [Figure5] as our pre-trained setting. The model is trained on ImageNet dataset for classifying images. It contains a convolution part and a fully connected part

which is used for classifying images.We have downloaded the complete VGG-16 architecture as we plan to use some of the fully-connected layers as well, but then later on we will remove the softmax layer as we don't need the classification layers, we are just planning to extract the feature representation of an image

Model: "vgg16"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0

Figure 5: pre-trained Model VGG-16

4.2 Attention based Architecture

We are using the Local attention based architecture for our model [4].Firstly, it produces the encoder hidden states, i,e encoder will produce hidden states for all the images in the input sequences. Then the alignment score is being calculated for each hidden state of the encoder and the decoder's previous hidden state. These score are then combined and softmax is applied on them. To generate the contextual information, the softmaxed scores and the encoder hidden states are then combined to formulate a vector representation. This vector is then combined to the last decoder hidden state and fed into the RNN to produce a new word respectively. This complete procedure is recursive in nature and the stopping criteria is till the length of caption generated surpasses the maximum length. To simplify and formulate generalized approach we followed the below mentioned steps -

- We extracted the features from the lower convolutional layer of VGG16 giving us a vector with 512 output channels.
- This vector is then passed through the CNN Encoder which consists of a single fully connected layer followed by a dropout layer.
- The Recurrent Neural Network(here GRU), the takes in the image to predict the next word.

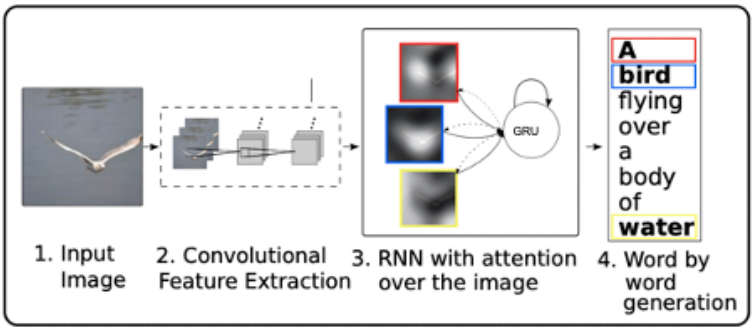

Figure 6: Attention based Architecture

- Furthermore, the attention based model enables us to see what parts of the image the model focuses on as it generates a caption.

4.3 Training Details

- The model architecture follows an encoder-decoder structure. Initially, the encoder output, the hidden state and the decoder output (<start> token at this stage) are passed to the decoder.
- The decoder returns output predictions and the hidden state.
- The hidden state is then fed back into the model to generate further predictions and loss is calculated.
- We make use of the attention property by passing the target word as the next input to the decoder. This helps the model learn correct sequence of word generation as well as improves the improves the training time. This is called the Teacher forcing technique, and it helps decide the next input of the decoder.
- We then calculate the gradient, apply it to the optimizer and perform backpropagation.
- Plotting a graph of the loss helps test against overfitting. This is especially useful if we use a smaller dataset.

4.4 Experiments Performed

- We ran 20 epochs over the training dataset which had 600 batches, as the model required high computation power given the size of the data and the size of the model, it took us 1300 seconds to run a single epoch.
- We have used Adam optimizer, along with Sparse Categorical Cross Entropy as our loss function.
- We have also plotted the loss over all the iterations , and observed that our model successfully converged all through the iterations and hence the training was successful as observed in the Fig.7.
- In order to evaluate the captions, we use a greedy approach based on Maximum Likelihood Estimation (MLE). We select the word which according to the model is most likely for the given input.
- During evaluation, the model performs similar to the training loop without the Teacher forcing method. The decoder input at each stage is the previous predictions, the hidden state and the encoder output.
- We use the BLEU measure to evaluate the result of the captions generated by the test set. The BLEU score is measured by taking the fraction of n-grams in the predicted sentence that appears in the ground-truth. It returns a value between 0 and 1, with a value closer to 1 meaning that they are very similar.
- We also define a function to plot the attention maps for each word generated.

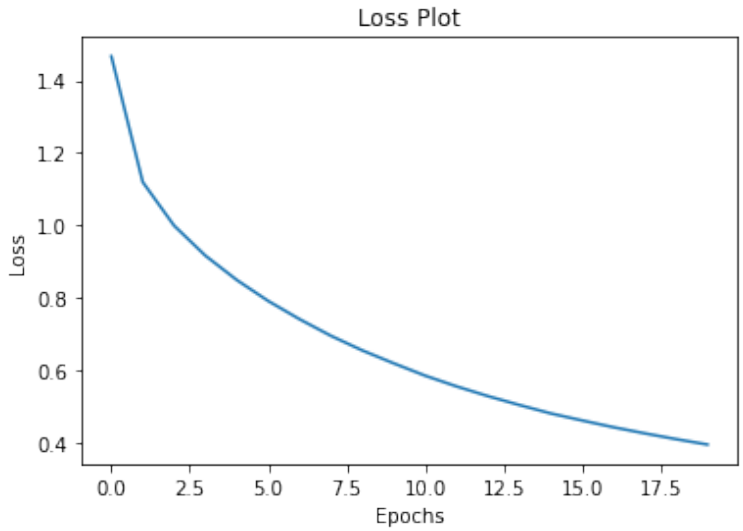

Figure 7: Plotting the Loss over 20 iterations

5 Evaluation

We are using greedy approach to evaluate the captions generated. The greedy approach computes the Maximum Likelihood Estimation (MLE) i.e.we select the word with the maximum logit value for a given output. We here greedily select the word which has the maximum probability.

5.1 Metric Used

We are using BLEU(Bilingual Evaluation Understudy) Score, and evaluating the generated captions on our test set [5]. The BLEU score, will take the fraction of the tokens present in the predicted sentence to the ones that appears in the ground truth. It return a value between 0 and 1. If the value is closer to one that means that the prediction is very close to the ground truth.

6 Observation/Results

After performing the experiments as mentioned above, we were able to get significant results. We have plotted attention plots and are also observing the predicted captions, with respect to the original caption. We are also monitoring the BLEU score for the test images. We ran our model on several test images and plotted the attention plot, so as to observe which part of the image was focused upon while predicting a particular word in a caption. In attention plot in figure 8 and 9 bottom image, we can observe that for every image the important feature is being highlighted and the predicted word is being also mentioned for that particular important feature. This clearly tells us that not every part of the image is important to predict a caption. Also, we can miss out on several minute details if we don't use this kind of attention architecture.
In figure 8, the ground truth caption was "Two white dogs are playing in snow". The predicted caption is "Two white dogs run across the snow". The reported BLEU score is 61.47, which is very significant. If we compare our results to the ground truth result which is around 75.

In figure 9, the ground truth caption was "Black dog with red collar is jumping in the water". The predicted caption is "black dog with red collar is jumping through the water" and the observed BLEU score is 67.32 which is also very significant. From the results of both the test images we observed that our model is able to reproduce significantly good captions. The captions are also semantically appropriate and they report outstanding BLEU scores.

Figure 8: Top: Ground Truth Caption "Two white dogs are playing in snow", Bottom: Attention Plot

7 Conclusion

The attention mechanism is highly utilized in recent years and is just the start of much more state of the art systems. Our trained model shows good performance on Flicker 8k dataset using the BLEU metric and the captions generated are interpretable and well aligned with human intuitions. We also understood that the images used for testing should be semantically very close to the ones used in the training images. We can also alter the evaluation method i.e we could use beam search in order to generate better captions. The attention model successfully captures the important features from an image and generates semantically sound captions as well. We can further work on its improvement so as to improve on the BLEU scores and predict more closer ground truth captions for an image.

8 Future Scope

In order to further improve the accuracy scores, we can try different things like:

- Use of the larger datasets, especially MS COCO dataset or the Stock3M dataset which is 26 times larger than MS COCO.
- Implement different attention mechanism like Adaptive Attention using Visual Sentinel and Semantic Attention [6].
- Implementing a Transformer based model which should perform much better than GRU.
- Implementing a better architecture for image feature extraction like Inception, Xception.
- We can do more hyperparameter tuning(learning rate, batch size, number of units, dropout rate) in order to generate better captions.
- We also want to address issues like model monitoring and interpretability using several different methods.

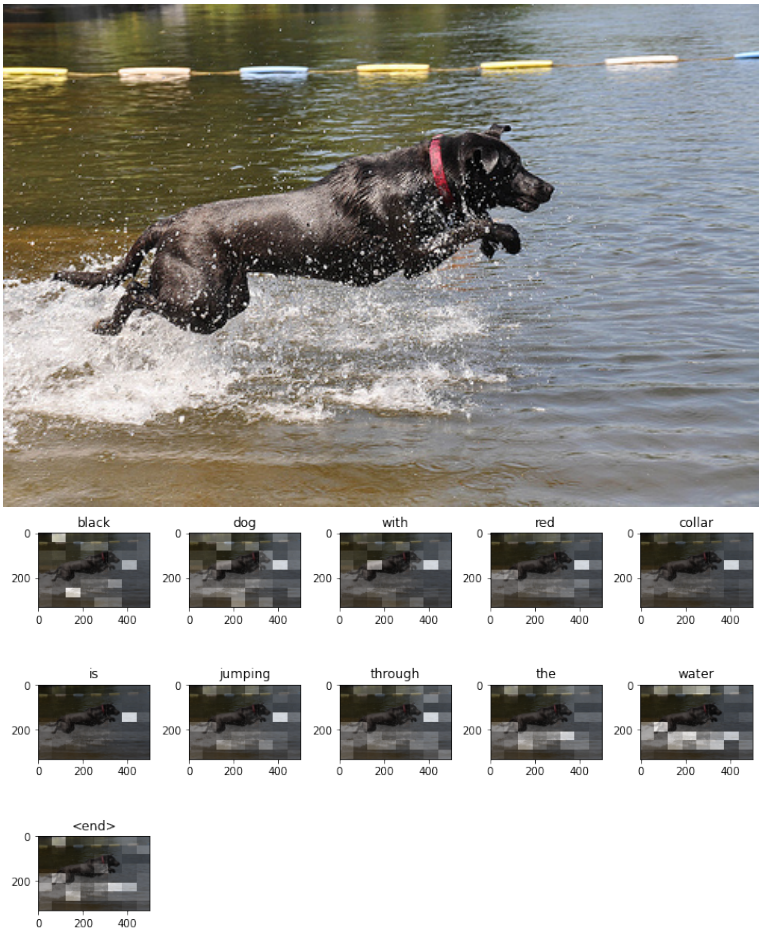

Figure 9: Top: Ground Truth Caption "Black dog with red collar is jumping in the water", Bottom: Attention Plot

9 Contributions

While the entire team collaborated in devising the problem statement, reading research papers, data gathering phase, architecture design and end conclusions, there were certain parts where each of us devoted extra time to ensure we remain on track and within the deadlines. We used Agile methodology for the project implementation wherein we divided our work equally and set up daily and weekly goals. We also had weekly meetings to discuss challenges and leanings, and daily-stand ups so as to keep a check on the status of the project. Overall, we divied our work as mentioned below -

- **Gourang Patel**: Architecture implementation and experimentations for enhanced caption performance.
- **Sanjan Vijayakumar**: Since the initial challenge was to get a good data set that can be solved keeping in line with the computational challenges, he devoted extra time in exploring datasets ranging from MS-COCO, Flicker30k and Flicker8k
- **Preet Shah and Sagar Singh**: Took the joint initiative to handle all the data preparation and preprocessing steps along with implementation aspects of Attention model exploring GRU and LSTM.

We further ensured to take part in weekly meetings to check on our respective progress and compile the work by the end of each week. This ensured that each of us took equal initiative by working in Agile phase and debugging roadblocks as encountered in the process of model training and development.

References

[1] Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. arXiv:1502.03044

[2] https://medium.com/swlh/image-captioning-using-attention-mechanism-f3d7fc96eb0e

[3] Image Captioning based on Deep Learning Methods: A Survey arXiv:1905.08110

[4] https://towardsdatascience.com/intuitive-understanding-of-attention-mechanism-in-deep-learning-6c9482aecf4f

[5] Improved Image Captioning via Policy Gradient optimization of SPIDEr arXiv:1612.00370

[6] Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning arXiv:1612.01887

[7] Refer this link for the implementation : https://github.com/Gourang97/attention-based-image-captioning