Neural Image Captioning with an Attention Based Mechanism

Shruti Verma
Department of Computer Science
University of Minnesota Twin Cities
Minnesota, MN 55455
verma125@umn.edu

Kirthi Kulkarni
Department of Electrical Engineering
University of Minnesota Twin Cities
Minnesota, MN 55455
kulka217@umn.edu

Debarati Das
Department of Computer Science
University of Minnesota Twin Cities
Minnesota, MN 55455
das00015@umn.edu

Nanda Unnikrishnan
Department of Electrical Engineering
University of Minnesota Twin Cities
Minnesota, MN 55455
unnik005@umn.edu

Kishor Kunal
Department of Electrical Engineering
University of Minnesota Twin Cities
Minnesota, MN 55455
kunal001@umn.edu

Abstract

In this project, we have attempted an implementation of image captioning from the work of Kelvin Xu et. al. "Show, attend and tell" [2]. This work implements the inception v3 CNN architecture in order to extract high level features and LSTM-RNN in order to learn the semantic of natural language in order to generate the related description. This includes an intermediate step called the attention layer which helps in learning when to prioritize the CNN learned feature or the LSTM-RNN word feature, based on the probability of the outputs. It differentiates words learnt through CNN or LSTM-RNN mainly for articles, prepositions, conjunctions, etc. Additionally, we implement a model with VGGNet which has lesser number of layers compared to inception. Also instead of using the LSTM cell we use a GRU-cell since it gives better performance. We have used MSCOCO dataset for this project.

1 Introduction

It goes without saying that the task of describing any image has a highly varied range of difficulty. Some images, such as a picture of a dog, an empty beach, or a bowl of fruit, may be on the easier end of the spectrum. While describing images of complex scenes which require specific contextual understanding—and to do this well, not just passably—proves to be a much greater captioning challenge. Providing contextual information to networks has long been both a sticking point, and a clear goal for researchers to strive for.

Image captioning interests us because it concerns what we understand about perception with respect to machines. For example, the same object ‘ flower ‘ throws images of different flowers in different people’s minds but the perceptions all hash around to the same word - “ flower”. This kind of

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

grounding of the conversation boils down to the language grounding problem. These ideas work with the ability to explain the results. If language grounding is achieved, then the network tells you how a decision was reached. In image captioning, a network is not only required to classify objects, but to describe objects (people and things) and their relationship in a given image. Hence, as we shall see, attention mechanisms and reinforcement learning are at the forefront of the latest advances—and their success may one day reduce some of the decision-process opacity that harms other areas of artificial intelligence research.

The most obvious image captioning application from which the reader benefits is from basic audio descriptions of images to plenty of opportunities to improve quality of life for the visually-impaired with annotations, real-time or otherwise. Largely, image captioning may benefit the area of retrieval, by allowing us to sort and request pictorial or image-based content in new ways. Mapping the space between images and language, in our estimation, may resonate with some deeper vein of progress. Which, once unearthed, could potentially lead to contextually-sophisticated machines. And, as we have noted before, providing contextual knowledge to machines may likely be one of the key pillars that eventually support AI's ability to understand and reason about the world as humans do.

Vision-to-Language problems present a particular challenge in Computer Vision because they require translation between two different forms of information. These kinds of problems encompass different domains such as object detection and Natural Language Processing. Image captioning systems currently focus on approaches which individually improve the detection as well as the sentence-creation system. Much of the recent progress in Vision-to-Language problems has been achieved through a combination of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). Our hypothesis involves individually introducing modifications/optimizations in the detection (CNN) as well as the sentence-creation system (RNN/LSTM). We also introduce an attention mechanism in the same system and observe the results. In this work, we discuss and demonstrate our experiments on Image Captioning systems, using the MS-COCO Dataset. We evaluate the quality of our captions using the BLEU, CIDEr and ROUGE scores.

2 Background

Image captioning, i.e. generating natural language description from visual data has been a difficult task due to its complexities in finding the relationship between the objects found through computer vision algorithms and finding relationship between these objects to create a semantically correct description. Various methods have been tried and tested so far in order to connect the feature extracted from image and find relationship through the semantics of a language. So far deep neural networks which use basic encoder and decoder architecture have paved the path for generating state of the art image captioning models. Our project is mainly based on the work of Oriol Vinyals et. al. [1] and Kelvin Xu et. al. [2], the first one is based on maximizing the probability of generating correct caption given an image and the latter has another module added to it called "attention" which helps in deciding when to select the CNN feature and when to select the text data which does not correspond to the image features extracted. In both the implementations, CNN extracts the image features and LSTM-RNN takes in both the features and the words of the captions. The semantics is learned from the captions word by word. The first paper discusses a neural probabilistic model by directly maximizing the probability of the correct translation given an input sentence in an "end-to-end" fashion – both for training and inference using the below formula:

$$\theta^* = argmax \sum_{(I,S)} P(S|I;\theta) \tag{1}$$

This model uses a recurrent neural network which encodes the variable length input into a fixed dimensional vector, and uses this representation to "decode" it to the desired output sentence. In the latter paper [2], they introduce an "attention" module. Two types of attention models are introduced, namely "hard" and "soft" attention. The features extracted are situated at different locations of the image, the algorithm tries to match certain words to the feature locations, it learns this with the help of backpropagation. We have implemented both aspects and produced certain comparisons with different CNN models.
The CNN(s) used in the above papers are VGGNet and Google's inception architecture to extract the high- level features. VGGNet consists of 16 convolutional layers and is very appealing because of its very uniform architecture. The Inception v3 model has 22 layers, we have implemented the VGGNet

from scratch on tensorflow but the also used the pre-trained weights of VGGNet and Inception v3 from the ImageNet challenge.
Coming to the NLP, we tokenize the captions into words and symbols such as ",", ".", etc. MSCOCO word exmbeddings were created with a vocabulary size of 5000 and vector size of 512, later we tested with a bigger vocabulary and word embedding which is more generalized called GloVe [3]. Global Vectors for Word Representation (GloVe) is provided by Stanford NLP team. Stanford provides various models from 25, 50 , 100, 200 to 300 dimensions base on 2, 6, 42, 840 billion tokens. Stanford NLP team apply word-word co-occurrence probability to build the embedding. In other word, if two words are co-exist many time, both words may have similar meaning so the matrix will be closer.

3 About the data

Dataset	
MS-COCO	Contains 120K images with 5 captions for each split : 80k images for Training and 40k images for Validation
Flickr8k	Contains 8K images with 5 captions each split : 7k images for training and 1k images for validation

3.1 Pre-processing Data

NLP-word embeddings
MSCOCO website has provided the images and a json file which has the mappings of the image ID to its corresponding captions. They have also provided a PythonAPI which we have used in our code in order to extract the data in the correct format. First, we extract the image ID(s) from the json file for all train, validation and test data. Next, the task was to create a vocabulary of the unique words words in the data along with their corresponding frequencies, also, we select the words which have a frequency above 5 occurrences. For these words we get the GloVe embeddings which represents words in a 300 dimensional space and the similar words would be close to each other in this 300-D space. A few other dictionaries are created for our convenience which store the mappings of captions to their respective ID(s) which will later help in mapping them to the images for train, test and validation datasets.
CNN-image embeddings
The images from MSCOCO were re-shaped into a 224×244 equally sized images, and were normalized on the RGB axis with respect to the $ilsvrc\_2012.npy$ file, from the ImageNet challenge. Along with this we also use the pre-trained Inception v3 model and further train it on MSCOCO dataset to get more concrete high level features.

4 Model

4.1 Overall architecture

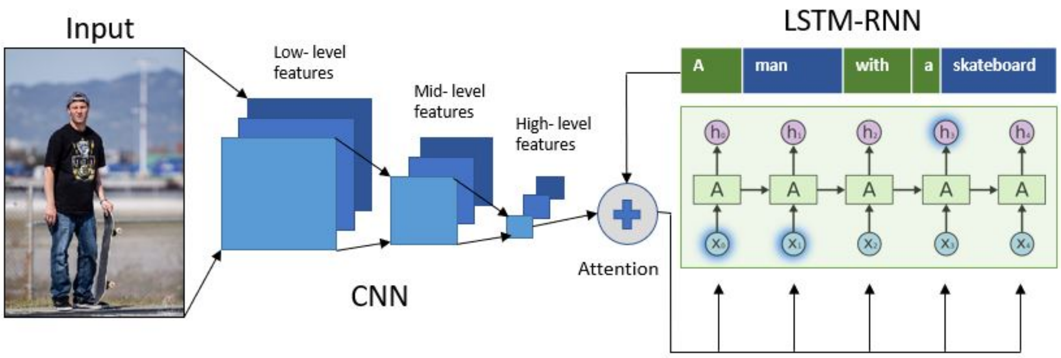


Figure 1: Overall model

The overall model has been implemented in Tensorflow. Initially the CNN extracts image features, it extracts a 300 dimension vector which represents these images. All similar images would be close to each other in this 300-D space. According to figure 1 above, The attention model has two inputs one is the image high-level features from CNN and then the caption words from the LSTM-RNN which are fed into the attention model.

4.2 Image feature extraction

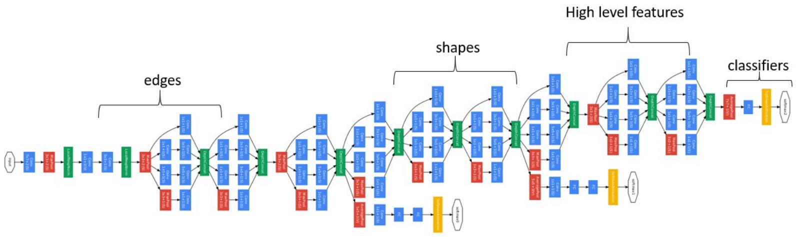


Figure 2: Inception model

A convolutional neural network can be used to create a dense feature vector. This dense vector, also called an embedding, can be used as feature input into other algorithms or networks. Here, we test the model with VGGNet and Inception v3 CNN architectures. For an image caption model, this embedding becomes a dense representation of the image and will be used as the initial state of the LSTM. We use 300 dimensional vector space in order to represent the image embedding.

4.3 Attention model

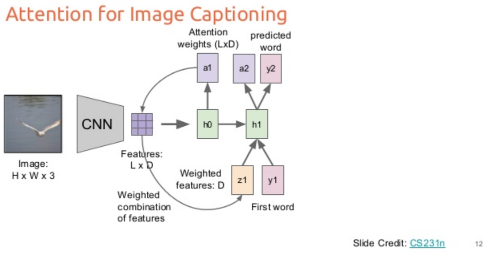


Figure 3: Attention model

Attention Mechanisms in Neural Networks are loosely based on the visual attention mechanism found in humans. It is related to human visual attention which essentially means to being able to focus on a certain region of an image with "high resolution" while perceiving the surrounding image in "low resolution", and then adjusting the focal point over time. With respect to image captioning, rather than compressing an entire image into a static representation, attention allows for salient features to dynamically come to the forefront as needed. This is especially important when there is a lot of clutter in an image.

The representations from convolutional network distills information in image down to most salient feature. In short it acts as a function encoding an image. The attention layer grabs a portion of these images, probabilistically determine the most important features and assign them higher weights. The recurrent network, a word generator picks up this weighted image feature vector, combines it with the previous state and generates words at every point in time. Learning stochastic attention requires sampling the attention location s_t each time, instead we can take the expectation of the context vector z_t directly, and formulate a deterministic attention model by computing a soft attention weighted annotation vector.

$$E_{p(s_t|a)[z_t]=\sum_{i=1} L\alpha_{t,i}a_i} \tag{2}$$

$$e_{ti} = f_{att}(a_i, h_{t-1}) \tag{3}$$

$$\alpha_{ti} = \frac{exp(e_{ti})}{\sum_{k=1}^{L} exp(e_{tk})} \tag{4}$$

4.4 LSTM- RNN

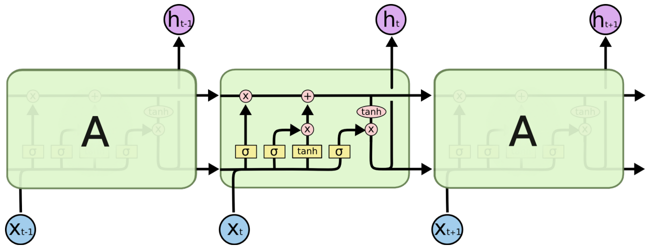


Figure 4: LSTM-RNN

An LSTM is a recurrent neural network architecture that is commonly used in problems with temporal dependencies. It succeeds in being able to capture information about previous states to better inform the current prediction through its memory cell state. An LSTM consists of three main components: a forget gate, input gate, and output gate. Each of these gates is responsible for altering updates to the cell's memory state.

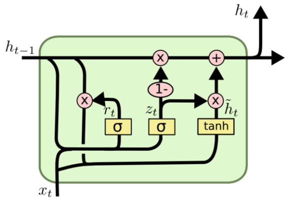


Figure 5: GRU cell

4.5 Beam Search

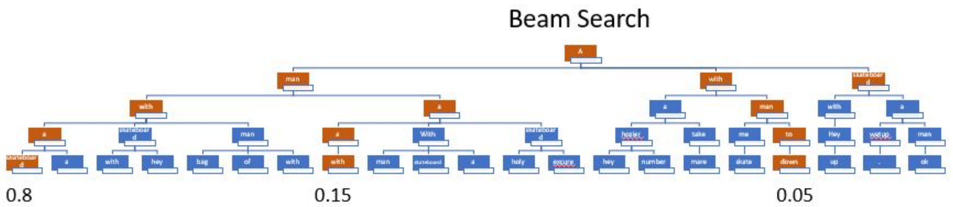


Figure 6: Beam Search

Once the model is generated and various probability values are assigned to each word and semantic relationship is built based on probability, beam search is used in order to predict the next word in the sentence given certain words have occurred. Beam search uses breadth-first search to build its search tree. At each level of the tree, it generates all successors of the states at the current level, sorting them in increasing order of heuristic cost. it only stores a predetermined number, β , of best states at each level (called the beam width). Here, we test our algorithm with beam size of 3 and 4. This is used for validation and test state once the model parameters are decided.

5 Experiments and Results

We tested our model with LSTM without attention, GRU without attention and LSTM with attention. for all the models, the dropout rate was set to 0.75, word and image embedding size was tested with

512 and 300 dimension vector, vocabulary size with 5000 and 9855 was tested, one layer in RNN, the model was tested with MSCOCO vocubulary embeddings and GloVe embeddings and a sparse softmax cross entropy loss function in tensorflow is used in order to calculate the loss with an Adam optimizer function from tensorflow.

1. Put results with lesser epochs and bad accuracy wala result
2. Final results and comparisons- LSTM vs GRU (hopefully) without attention, VGG and inception

Model	BLEU-1	METEOR	CIDEr	ROUGE$_L$	epochs	loss
LSTM	0.701	0.231	0.896	0.506	49	2.454
GRU	0.706	0.231	0.900	0.508	24	2.19
LSTM attention	0.59	0.18	0.463	0.446	30	4.5

As observed above this model performs really well with the GRU cell for lesser number of epochs and we got better results on GRU- RNN model compared to LSTM- RNN.

Loss and results for LSTM-RNN model:

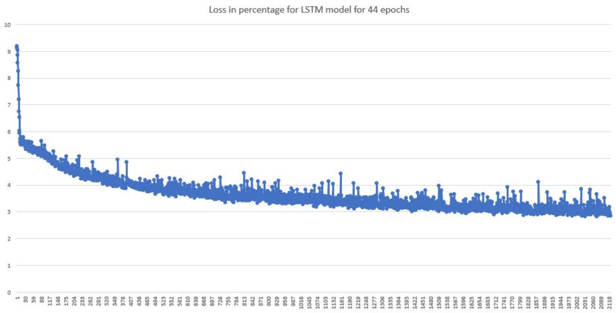
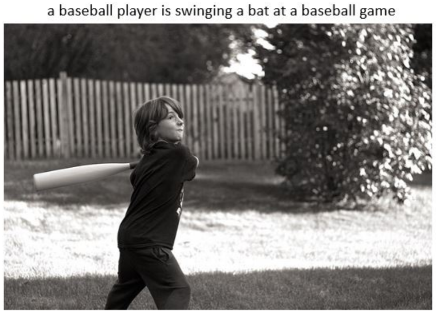
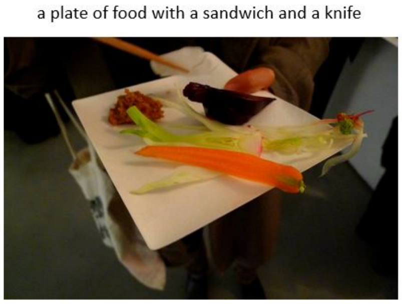


Figure 7: Total loss over all batches for LSTM without attention (total loss= 2.454%)





Loss and results for GRU-RNN model:

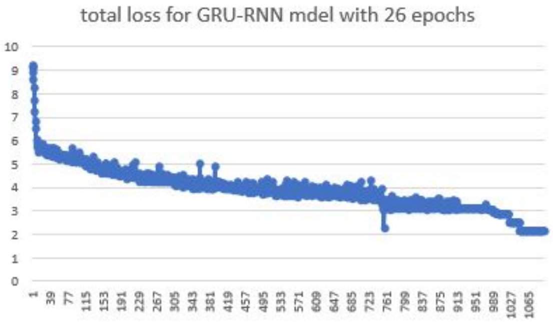
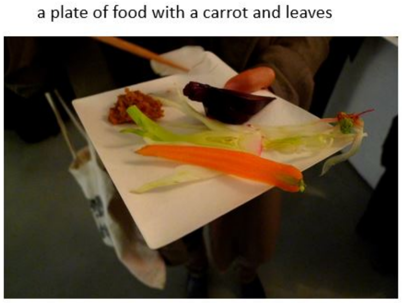


Figure 6: Total loss over all batches for GRU- RNN without attention (total loss= 2.10%)

A street with a sign


a plate of food with a carrot and leaves

Loss and results for LSTM with attention model:

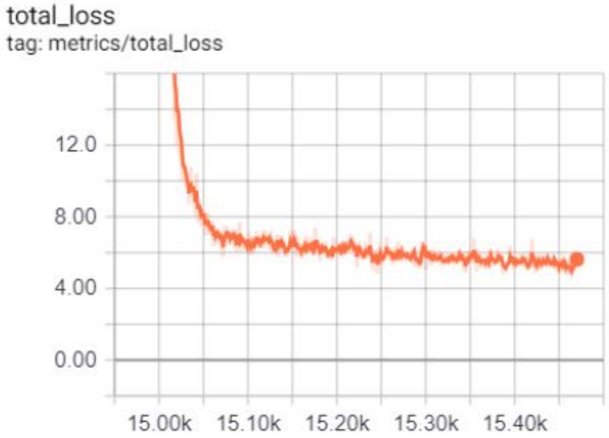
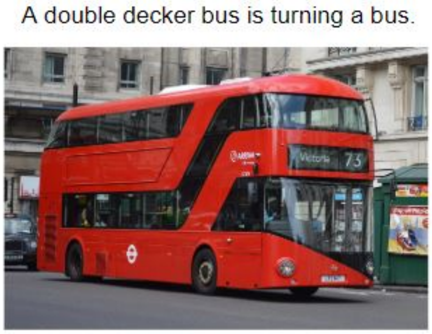


Figure 6: Total loss over all batches for LSTM attention (total loss= 4.5%)


A double decker bus is turning a bus.

A man riding a surfboard riding a surfboard.

6 Conclusion

Through this project we learnt the inner workings and complexities of Tensorflow, handling huge amounts of data and working of RNN and CNN with image and text data. We conclude that GRU gave us much better results and the execution was faster. For huge amounts of data GRU cell had an impact on the speed and almost equivalent performance with respect to accuracy (better by a slight margin in our case). We could not receive an improvement for LSTM-RNN with the attention model in our project.

7 Future Work

Some of the thing we would like to improvise or try with our current model were:
1. Training CNN and LSTM separately on same data but different labels. Connecting these trained

layers together to achieve minimum loss on the combined layer.
2. Implement Contrastive Learning to the image captioning model to maintain distinctive in the generated captions.
3. Implementing the captioning process in two stage:
a) extracting an explicit semantic representation from the given image,
b) constructing the caption based on a recursive compositional procedure in a bottom-up manner.
4. Image captioning as a conditional GAN training, proposing both a context-aware LSTM captioner and co-attentive discriminator, which enforces semantic alignment between images and captions.

8 References

1. Show and tell: A neural image caption generator, Oriol Vinayals et. al. https://arxiv.org/pdf/1411.4555.pdf 2. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, Kelvin Xu et. al; https://arxiv.org/pdf/1502.03044.pdf
3. Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. https://nlp.stanford.edu/pubs/glove.pdf
4. Rethinking the Inception Architecture for Computer Vision, https://arxiv.org/abs/1512.00567
5. http://cvlab.postech.ac.kr/research/text_att
6. http://kelvinxu.github.io/projects/capgen.html
7. Improved Image Captioning with Adversarial Semantic Alignment, Pierre L. Dognin, Igor Melnyk, Youssef Mroueh, Jarret Ross, Tom Sercu (IBM Research, USA) https://arxiv.org/pdf/1805.00063.pdf
8. https://cs.stanford.edu/people/karpathy/sfmltalk.pdf
9. http://www.wildml.com/2016/01/attention-and-memory-in-deep-learning-and-nlp/
10. Contrastive Learning for Image Captioning, Bo Dai, Dahua Lin https://papers.nips.cc/paper/6691-contrastive-learning-for-image-captioning.pdf
11. A Neural Compositional Paradigm for Image Captioning, Bo Dai, Sanja Fidler, Dahua Lin