# Social Media Image Captioning using Attention Mechanism

**Title:** Image Captioning using Attention Mechanism

**Author(s):** Poornapragna Vadiraj, Varun Bhaseen, Mirsaeid Abolghasemi

**Abstract**: The main idea of this project is to find a good related caption for images on social media. Most of the images social media platforms have no captions or not related captions. So, this application can help to put a caption for those images for further usages. In this application users can enter the image and get a related caption based on the content on the image.

**Introduction:**

Attention mechanisms are broadly used in present image captioning encoder / decoder frameworks, where at each step a weighted average is generated on encoded vectors to direct the process of caption decoding. However, the decoder has no knowledge of whether or how well the vector being attended and the attention question being given are related, which may result in the decoder providing erroneous results. Image captioning, that is to say generating natural automatic descriptions of language images are useful for visually impaired images and for the quest of natural language related pictures. It is significantly more demanding than traditional vision tasks recognition of objects and classification of images for two guidelines. First, well formed structured output space natural language sentences are considerably more challenging than just a set of class labels to predict. Secondly, this dynamic output space enables a more thin understanding of the visual scenario, and therefore also a more informative one visual scene analysis to do well on this task.

**Related Work:**

Image captioning has been extensively studied recently with encoder-decoder versions, see e.g. [1, 2, 3, 4, 5]. A CNN processes the input image in its basic form to transform it into a vector representation which is used as the initial input for an RNN. Given the previous word, the RNN sequentially predicts the next word in the caption without limiting the temporal dependence to a fixed order, as in n-gram-based approaches. The representation of the CNN image can be entered in different ways in the RNN.While some authors [6, 7] Use this only to calculate the initial RNN status, while others enter it in each RNN iteration [8, 9].

**Data:**

To address this problem, there are many open source datasets available, such as Flickr 8k (containing 8k images), Flickr 30k (containing 30k images), MS COCO (containing 180k images), etc. But we have used the Flickr 8k dataset for this case study which you can access from here. Training a model with a large number of images on a system that is not a very high end PC / Laptop may not be feasible either. This dataset contains 8000 images with 5 captions each (as we already saw in the Introduction section that an image can have multiple captions, all of which are relevant at the same time). These images are bifurcated as follows: Training Set — 6000 images, Dev Set — 1000 images, and Test Set — 1000 images.

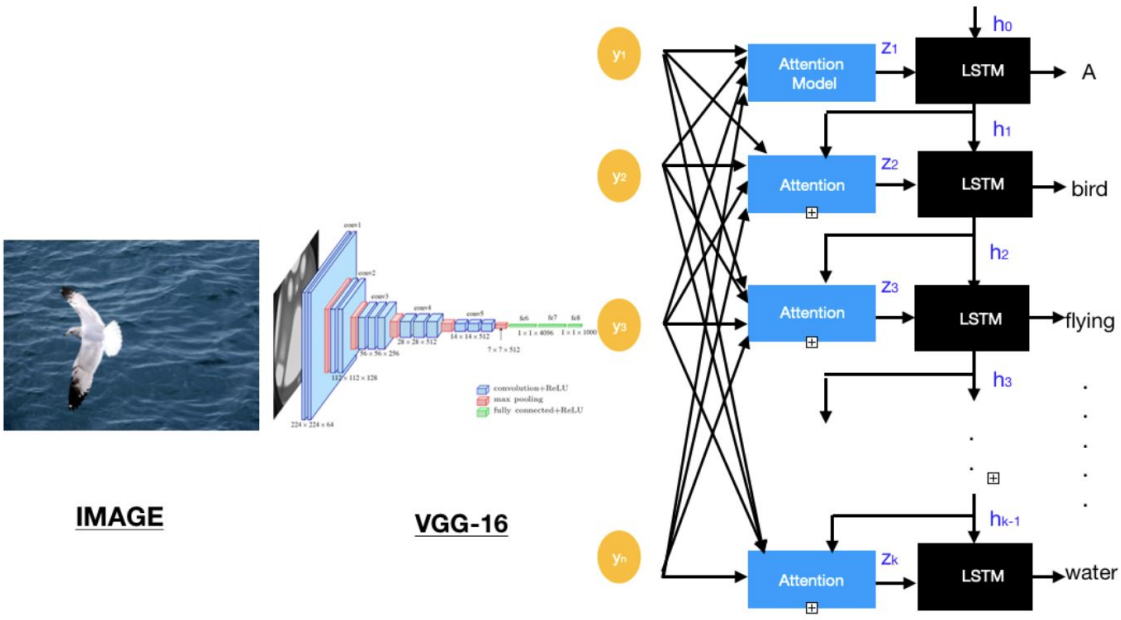
**Methods:**

A "classic" image captioning system will encode the image, using a pre-trained Convolutional Neural Network (ENCODER) to produce a hidden state. Then, by using an LSTM(DECODER) it would decode this hidden state, and generate each word of the caption recursively. Deep learning methods demonstrated state-of-the-art results on problems related to the generation of captions. What is most remarkable about these approaches is that it is possible to define a single end-to - end model to predict a caption, provided a picture, rather than involve complex data preparation or a pipeline of explicitly constructed models.

The last layer of the CNN is usually the softmax layer, which assigns the likelihood that every entity may be in the picture. However, if we strip the softmax layer from CNN, we can feed the rich encoding of the picture by CNN into the DECODER (language generation of RNN) built to generate sentences.

We may then specifically train the entire program on images and their captions, thereby increasing the probability that the explanations it generates can better fit the explanations of each image for the testing. An NPY file is a NumPy array file generated with NumPy library enabled by the Python software package. It includes an array stored in file format NumPy (NPY). NPY files hold all the details required to recreate an array on any device, including knowledge about the sort and the design. Objects from TensorFlow have a simple automated process to save and restore the values of the variables they use.

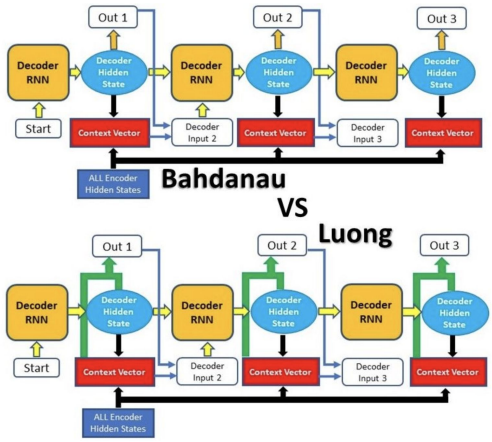
* Both prefixes are clustered into a single checkpoint file where the CheckpointManager stores the state.




Attention Mechanism used in Image Captioning

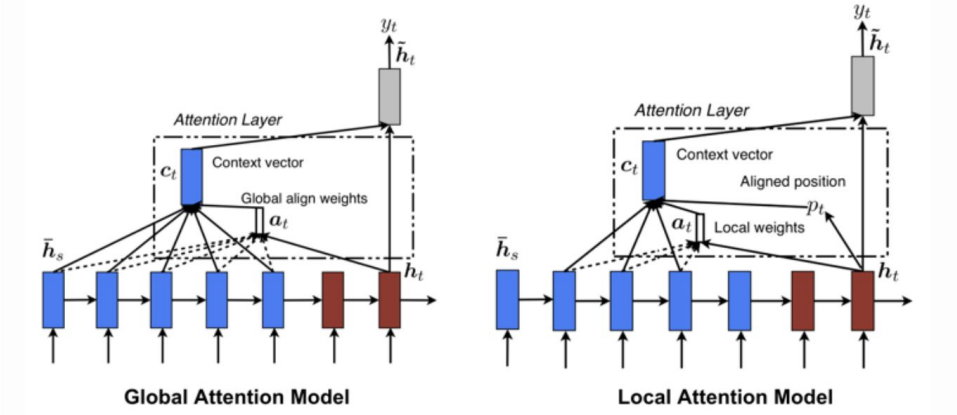
**Types of Attention Mechanism:**

Attention could be broadly differentiated into 2 types:

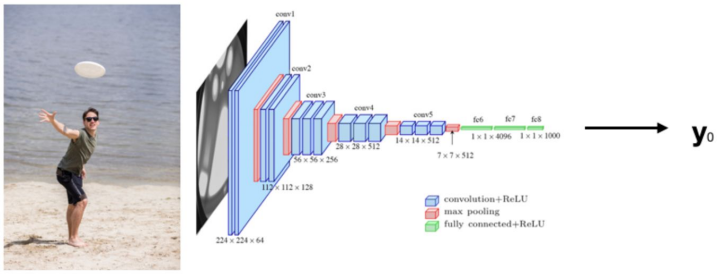



Global Attention(Luong's Attention): Attention is placed on all source positions.

Local Attention(Bahdanau Attention): Attention is placed only on a few source positions.




Global (Luong's Attention) vs Local(Bahdanau Attention) Attention mechanisms




The code creates an instance of the VGG16 model using the Keras API. This automatically downloads the required files if you don't have them already.

The VGG16 model was pre-trained on the ImageNet data-set for classifying images. The VGG16 model contains a convolutional part and a fully-connected (or dense) part which is used for the image classification.

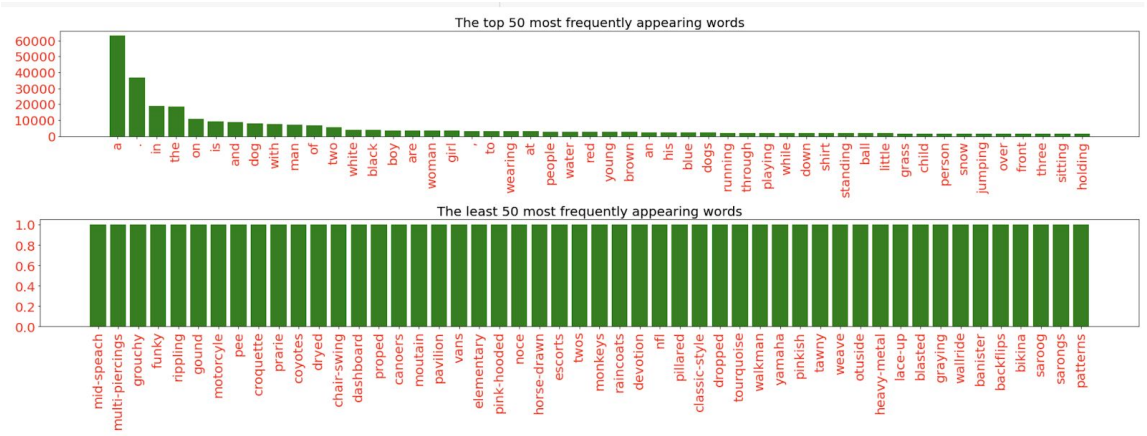
If include_top=True then the whole VGG16 model is downloaded which is about 528 MB. If include_top=False then only the convolutional part of the VGG16 model is downloaded which is just 57 MB.

We will use some of the fully-connected layers in this pre-trained model, so we have to download the full model, but if you have a slow internet connection, then you can try and modify the code below to use the smaller pre-trained model without the classification layers.

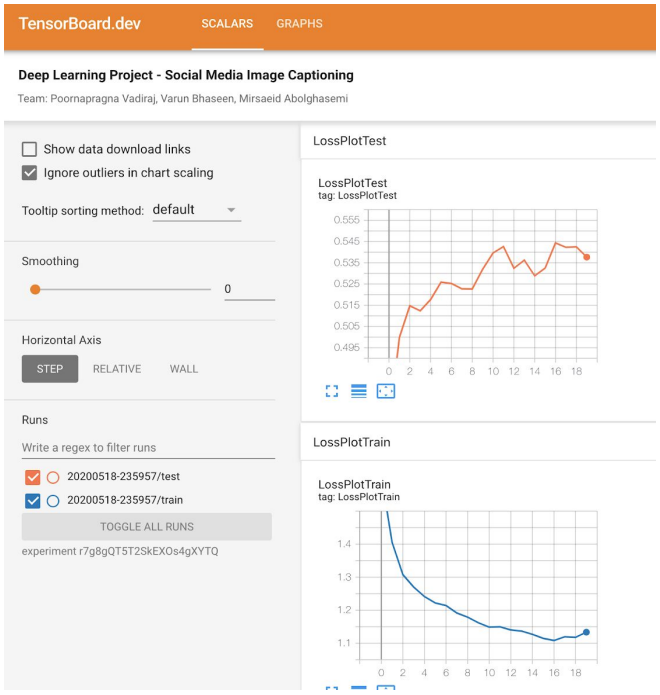
Tutorials #08 and #10 explain more details about Transfer Learning.

## Experiments:

The problem with this method is that, when the model attempts to generate the caption next word, that word usually only describes a part of the image. It is not capable of capturing the essence of the whole input image. Using the entire representation of image h to condition the generation of each word can not generate different words effectively for different parts of the image. This is why an Attention mechanism can be helpful.




## Tensorboard Logs :



**Conclusion:**

Over all, we have to admit that our simple first-cut model does a good job of producing captions for pictures, without any stringent hyper-parameter tuning.We have to recognize that the photos used for research have to be semantically linked to those used in model training. For eg, if we train our model on the photos of cats, dogs, etc., we don't have to check it on airplane photographs, waterfalls, etc. This is an example where the distribution of the train and test sets will be very different and no Machine Learning model will deliver good performance in the world in such cases. The result for Beam Search was better generated than Greedy Search.

**Supplementary Material**:

Our supplementary material might include:

- Source code (if your project proposed an algorithm, or code that is relevant and important for your project.).
- Cool videos, interactive visualizations, demos, etc.

**References:**

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In ECCV, 2016.

[2] 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 visual question answering. In CVPR, 2018.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. arXiv preprint arXiv:1607.06450, 2016.

[4] Hedi Ben-younes, Remi Cadene, Matthieu Cord, and Nicolas Thome. Mutan: Multimodal tucker fusion for visual question answering. In ICCV, 2017.

[5] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In NeurIPS, 2015.

[6] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In CVPR, 2017.

[7] Yangyu Chen, Shuhui Wang, Weigang Zhang, and Qingming Huang. Less is more: Picking informative frames for video captioning. In ECCV, 2018.

[8] Marco Pedersoli and Thomas Lucas and Cordelia Schmid and Jakob Verbeek: Areas of Attention for Image Captioning, arXiv, 2016.

[9] Sen He, Wentong Liao, Hamed R. Tavakoli, Michael Yang, Bodo Rosenhahn, and Nicolas Pugeault: Image Captioning through Image Transformer, arXiv, 2020