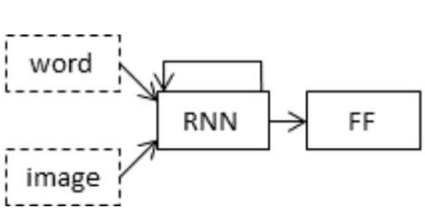
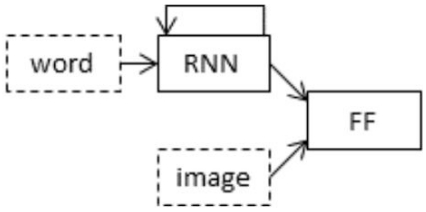
Where to put the Image in an Image Caption Generator

When a recurrent neural network language model is used for caption generation, the image information can be fed to the neural network either by directly incorporating it in the RNN – conditioning the language model by '**injecting**' image features – or in a layer following the RNN – conditioning the language model by '**merging**' image features. Injecting and merging models are presented in Figure 1.



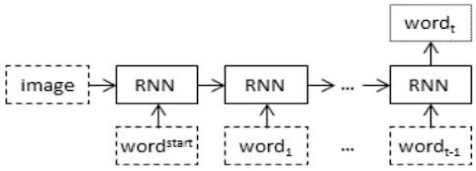
(a) Conditioning by injecting the image means injecting the image into the same RNN that processes the words.



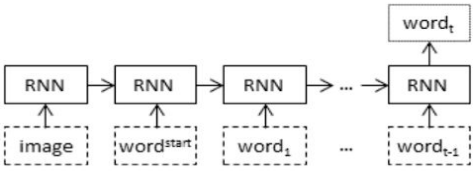
(b) Conditioning by merging the image means merging the image with the output of the RNN after processing the words.

Figure 1. Injecting and Merging architectures.

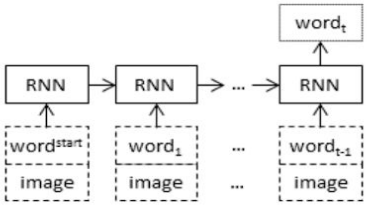
The difference between those architectures is that in injecting architecture, the image feature vector is given as an input to RNN except word tokens, in turn, in the merge model, the image feature vector is introduced to the Feed Forward network together the output of the RNN network. The injecting architecture is also split to three different ways that is described in Figure 2.



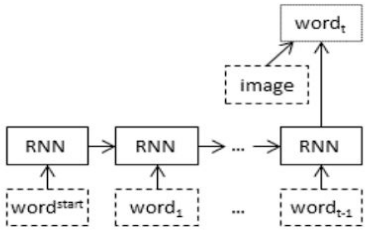
(a) Init-inject: The image vector is used as an initial hidden state vector for the RNN.



(b) Pre-inject: The image vector is used as a first word in the prefix.



(c) Par-inject: The RNN accepts two inputs at once in every time step: a word and an image.



(d) Merge: The image vector is merged with the prefix outside of the RNN.

Figure 2. Different ways of conditioning a neural language model with an image

In this project, injecting and merging models were implemented on Flickr 8K dataset and evaluated using bleu scores.
Dataset: Flickr8K is used as a dataset for this project. It has own splits for the train (6K), dev (1K) and test (1K) sets. In the dataset, five different sentence descriptions are presented for each image. The following configurations were applied on the sentences:

- Remove punctuation from each token
- Remove tokens with numbers in them
- Tokenization

As an encoder, VGG pretrained network was used to get the image feature vector. The preprocessing steps for the images are given as follows:

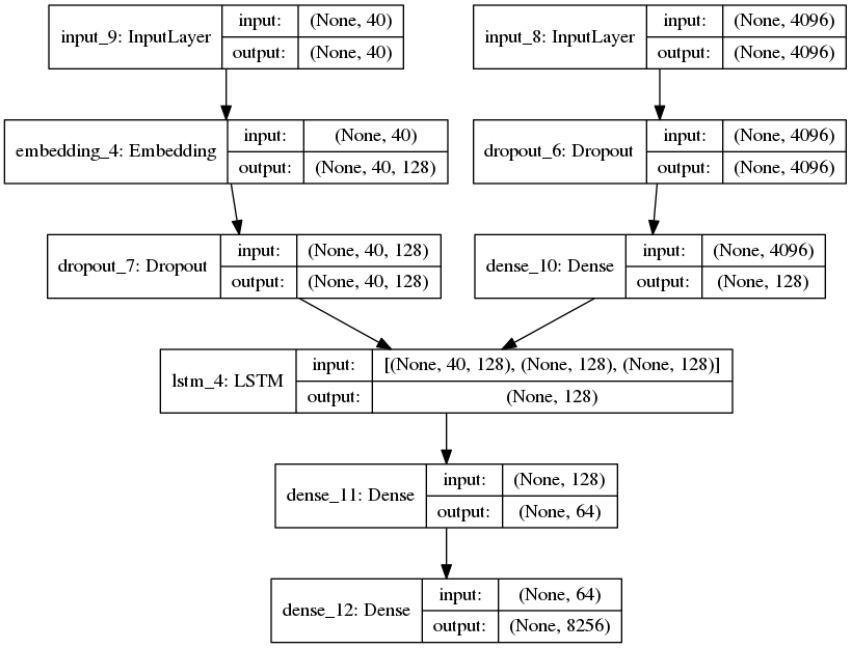
- Remove the last layer from the model
- Get the image features which contains 4096 element vector
- Normalize the image

As a decoder LSTM-based recurrent neural network was used. The embedding size was chosen as the size of the vocabulary which is 8256. The sequences were padd with the maximum length of the sentence length which is 40.

Methods:

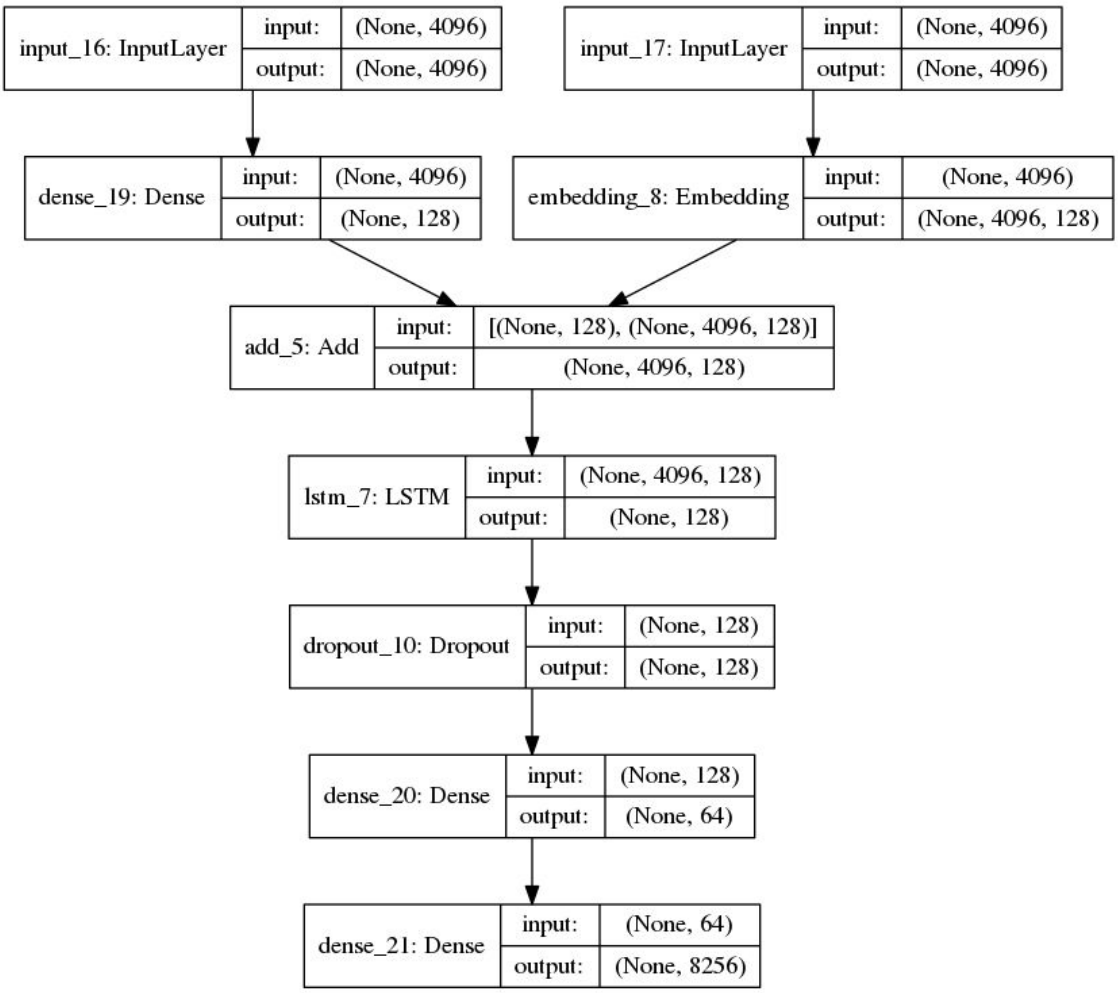
Init-inject model:

The image feature vector is given to the LSTM network as hidden states.



Pre-inject model:

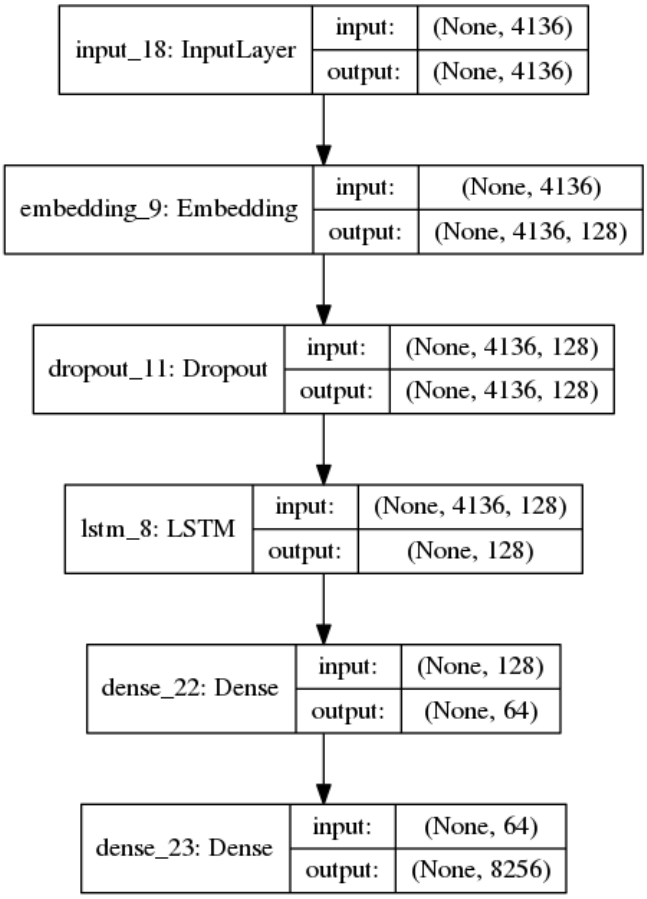
The image vector is used as the first word prefix. To introduce the image vector and the word sequences to the LSTM network, the word sequence vector is padded with the length of the image feature vector.



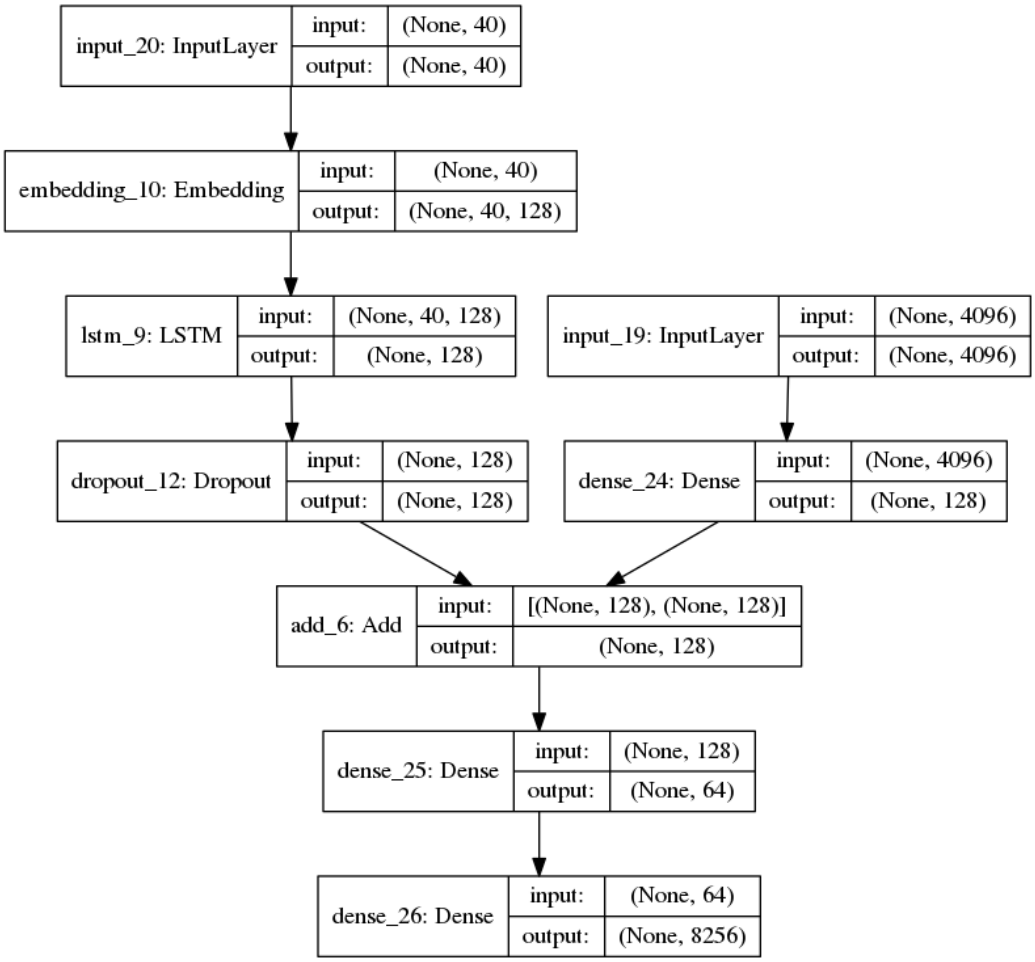
Par-inject model:

In this model, the image vector is given to the LSTM network together with the word

sequences at each time step. That is why, I concatenated the image and word sequence vector before feeding to the LSTM network.



Merge model: In this model, the image feature vector is merged with prefix outside of RNN.



The evaluation of the models with bleu score:
Models were trained for 1 epoch

Methods	Bleu-1	Bleu-2	Bleu-3	Bleu-4
init-inject	**0.303877**	0.118964	0.063030	0.022465
pre-inject	0.283756	0.083527	0.031183	0.011743
par-inject	0.238469	0.092813	0.030492	0.014862
merge	0.294013	**0.123476**	**0.075956**	**0.028844**

As the training for 1 epoch took more than 12 hours, there was not enough time for bigger epochs. As we see from the results the merge model outperforms the other

model in most of the cases. I chose the merged model and trained for 5 epochs. The predictions for some images as following:

Predictions:
Actual:
A dog runs across a grassy lawn near some flowers .
A brown dog running over grass .
A yellow dog is playing in a grassy area near flowers .
A brown dog with its front paws off the ground on a grassy surface near red and purple flowers .
A brown dog running

Predicted:
A black and white dog is running through a field .

Actual:
A surfer in all black is riding a wave .
A woman in a black wetsuit surfs in bad weather .
A surfer rides the waves .
A person is surfing .
A person in a black wetsuit is surfing in the ocean with a wave coming down .

Predicted:
A surfer is riding a wave .

As an conclusion, we could not say for one epoch results that which model is good or bad. However, in the paper, It is also confirmed that it is not especially detrimental to compare whether one architecture is better or worse. It is concluded that the merge model gives less descriptive results, in turn, the inject models give more descriptive but short sentences. As inject models use bigger input for the LSTM network, It can be reason that the output gives more descriptive results. However, as the size of the inject models is bigger than the merge model, the training process is quite time consuming.