



Description Generation from Images:

Proposed model pipeline using Python and Keras

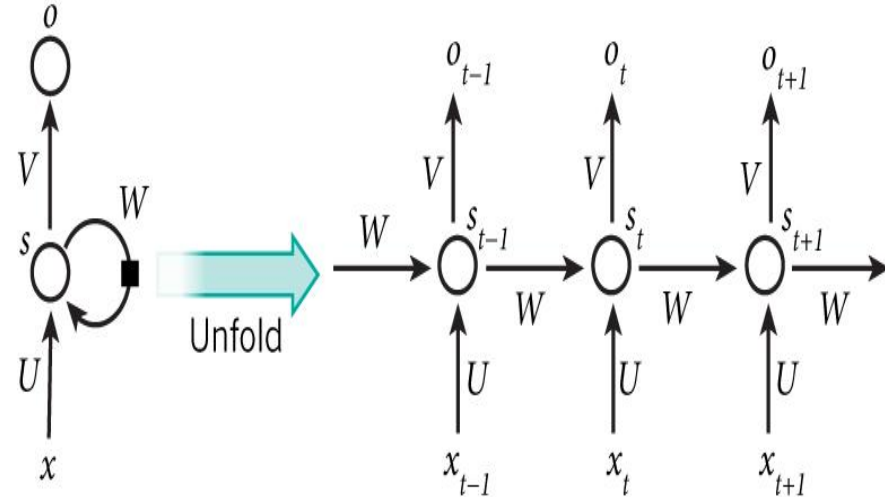
Anup Deshmukh
IMT2014013

Advised by: Prof J. Dinesh

Sections: Implementation

- Flickr8 dataset
- Generating Image Representation ✓
- Generating Text Representation ✓
- Loading the Data ✓
- Defining the Encoder-Decoder LSTM model
- Fitting the Enc-Dec LSTM model on both representations
- Model Evaluation

✓ Denotes implemented





1.1 FLicker8 Dataset

1

Flickr8k.token.txt - the raw captions of the Flickr8k Dataset . The first column is the ID of the caption which is "image address # caption number"

2

Flickr8k.lemma.txt - the lemmatized version of the above captions

3

Flickr_8k.trainImages.txt - The training images, **Flickr_8k.devImages.txt** - The development/validation images, **Flickr_8k.testImages.txt** - The test images

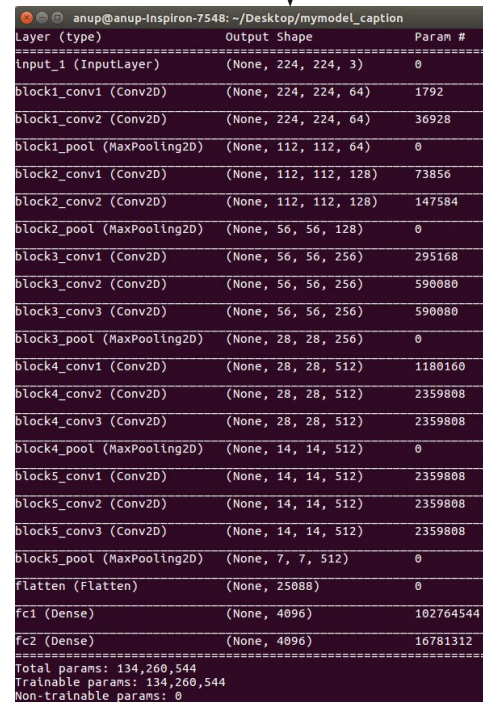
4

ExpertAnnotations.txt is the expert judgments. The first two columns are the image and caption IDs. Caption IDs are <image file name>#<0-4>. Scores range from 1 to 4 where 4 meaning the good caption.

1.2 Generating Image Representation

- 1 Pretrained weights of VGG net (CNN) model. Input: Flicker8k_Dataset Images
- 2 Convert raw image into NumPy array and pre-process the input which is suitable for VGG net.
- 3 The image features are 1-dimensional 4,096 element vector.

Architecture of used VGG net

A terminal window screenshot showing the architecture of a VGG net. The terminal output lists layers, their types, output shapes, and the number of parameters. The layers include input, convolutional, pooling, and fully connected layers. The total number of parameters is 134,260,544, with 134,260,544 trainable parameters and 0 non-trainable parameters.

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)	590880
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590880
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
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
Total params: 134,260,544		
Trainable params: 134,260,544		
Non-trainable params: 0		

1.3 Generating Text Representation

Generated Dictionary

1

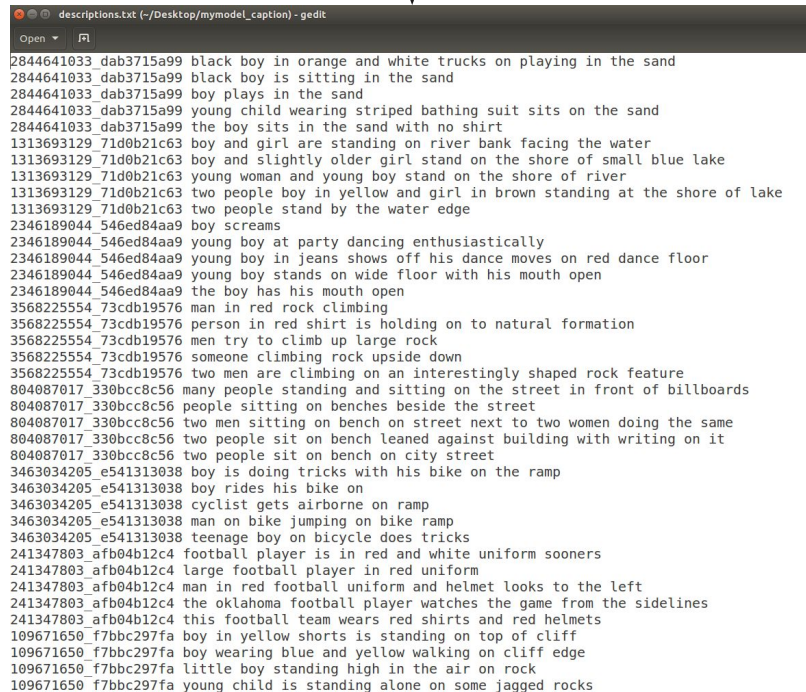
Load raw training data of the format: #id
#raw_caption Input:
Flickr8k_text/Flickr8k.token.txt

2

Create the dictionary of the form shown
in Image.

3

Clean the dictionary by: Removing
punctuations, fillers, numbers if any. Also
create the vocabulary.



```
descriptions.txt (-/Desktop/mymodel_caption) - gedit
Open
2844641033 dab3715a99 black boy in orange and white trucks on playing in the sand
2844641033 dab3715a99 black boy is sitting in the sand
2844641033 dab3715a99 boy plays in the sand
2844641033 dab3715a99 young child wearing striped bathing suit sits on the sand
2844641033 dab3715a99 the boy sits in the sand with no shirt
1313693129 71d0b21c63 boy and girl are standing on river bank facing the water
1313693129 71d0b21c63 boy and slightly older girl stand on the shore of small blue lake
1313693129 71d0b21c63 young woman and young boy stand on the shore of river
1313693129 71d0b21c63 two people boy in yellow and girl in brown standing at the shore of lake
1313693129 71d0b21c63 two people stand by the water edge
2346189044 546ed84aa9 boy screams
2346189044 546ed84aa9 young boy at party dancing enthusiastically
2346189044 546ed84aa9 young boy in jeans shows off his dance moves on red dance floor
2346189044 546ed84aa9 young boy stands on wide floor with his mouth open
2346189044 546ed84aa9 the boy has his mouth open
3568225554 73cdb19576 man in red rock climbing
3568225554 73cdb19576 person in red shirt is holding on to natural formation
3568225554 73cdb19576 men try to climb up large rock
3568225554 73cdb19576 someone climbing rock upside down
3568225554 73cdb19576 two men are climbing on an interestingly shaped rock feature
804087017 330bcc8c56 many people standing and sitting on the street in front of billboards
804087017 330bcc8c56 people sitting on benches beside the street
804087017 330bcc8c56 two men sitting on bench on street next to two women doing the same
804087017 330bcc8c56 two people sit on bench leaned against building with writing on it
804087017 330bcc8c56 two people sit on bench on city street
3463034205 e541313038 boy is doing tricks with his bike on the ramp
3463034205 e541313038 boy rides his bike on
3463034205 e541313038 cyclist gets airborne on ramp
3463034205 e541313038 man on bike jumping on bike ramp
3463034205 e541313038 teenage boy on bicycle does tricks
241347803 afb04b12c4 football player is in red and white uniform sooners
241347803 afb04b12c4 large football player in red uniform
241347803 afb04b12c4 man in red football uniform and helmet looks to the left
241347803 afb04b12c4 the oklahoma football player watches the game from the sidelines
241347803 afb04b12c4 this football team wears red shirts and red helmets
109671650 f7bbc297fa boy in yellow shorts is standing on top of cliff
109671650 f7bbc297fa boy wearing blue and yellow walking on cliff edge
109671650 f7bbc297fa little boy standing high in the air on rock
109671650 f7bbc297fa young child is standing alone on some jagged rocks
```



1.4.1 Loading the Data

1

Each description will be split into words. The model will be provided one word and the photo and generate the next word. Then the first two words of the description will be provided to the model as input with the image to generate the next word. This is how the model will be trained.

X1	X2 (text sequence)	y (word)
photo	startseq	sachin
photo	startseq, sachin,	is
photo	startseq, sachin, is	tired
photo	startseq, sachin, is, tired	endseq



1.4.2 Loading the Data

2

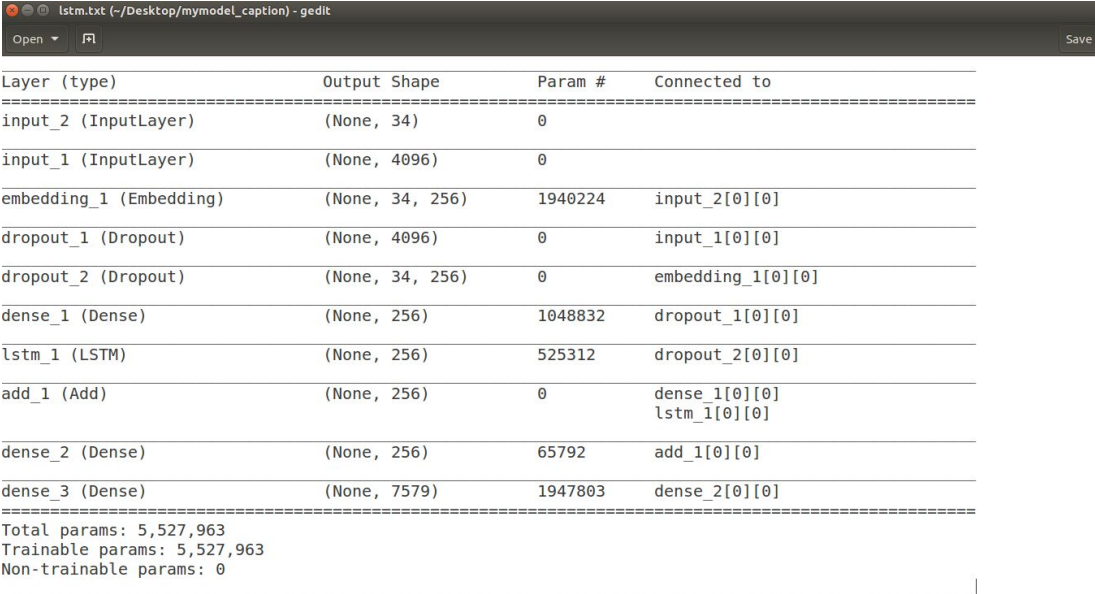
The input text is encoded as integers, which will be fed to a word embedding layer. The photo features will be fed directly to another part of the model. The model will output a prediction, which will be a probability distribution over all words in the vocabulary.

3

The output data will therefore be a one-hot encoded version of each word, representing an idealized probability distribution with 0 values at all word positions except the actual word position, which has a value of 1.

1.5.1 Defining the Encoder-Decoder LSTM model

1



The screenshot shows a Keras model summary for an Encoder-Decoder LSTM model. The model is defined in a file named 'lstm.txt' located at '~/Desktop/mymodel_caption'. The summary table lists the layers, their types, output shapes, parameter counts, and connections. The layers include two input layers, three embedding/dropout/dense layers, and one LSTM layer. The total number of parameters is 5,527,963, with 5,527,963 trainable parameters and 0 non-trainable parameters.

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	(None, 34)	0	
input_1 (InputLayer)	(None, 4096)	0	
embedding_1 (Embedding)	(None, 34, 256)	1940224	input_2[0][0]
dropout_1 (Dropout)	(None, 4096)	0	input_1[0][0]
dropout_2 (Dropout)	(None, 34, 256)	0	embedding_1[0][0]
dense_1 (Dense)	(None, 256)	1048832	dropout_1[0][0]
lstm_1 (LSTM)	(None, 256)	525312	dropout_2[0][0]
add_1 (Add)	(None, 256)	0	dense_1[0][0] lstm_1[0][0]
dense_2 (Dense)	(None, 256)	65792	add_1[0][0]
dense_3 (Dense)	(None, 7579)	1947803	dense_2[0][0]

Total params: 5,527,963
Trainable params: 5,527,963
Non-trainable params: 0



1.5.2 Defining the Encoder-Decoder LSTM model

1

Photo Feature Extractor. This is a 16-layer VGG model pre-trained on the ImageNet dataset. We have pre-processed the photos with the VGG model (without the output layer) and will use the extracted features predicted by this model as input.

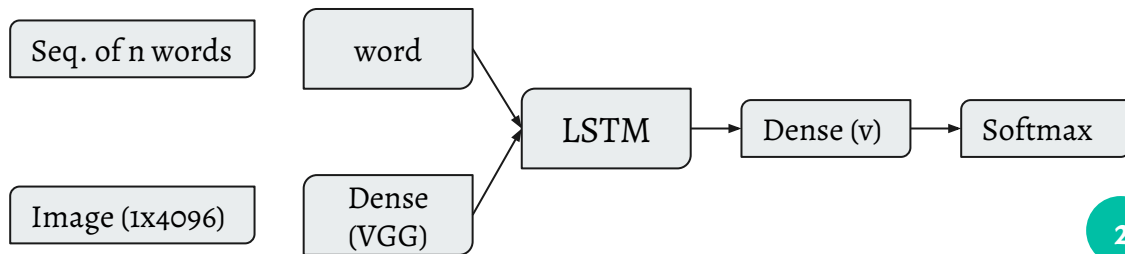
2

Sequence Processor: This is a word embedding layer for handling the text input, followed by a Long Short-Term Memory (LSTM) recurrent neural network layer. *Discussed in previous slide

3

Decoder: Both the feature extractor and sequence processor output a fixed-length vector. These are merged together and processed by a Dense layer to make a final prediction.

1.6.1 Fitting the Enc-Dec LSTM model on both representations



Architecture 1: Inject

'Dense' means fully connected layer with bias and v is the vocab size

1

The key property of these models is that the CNN image features are used to condition the predictions of the best caption to describe the image. However, this can be done in different ways and the role of the RNN depends in large measure on the mode in which CNN and LSTM are combined.

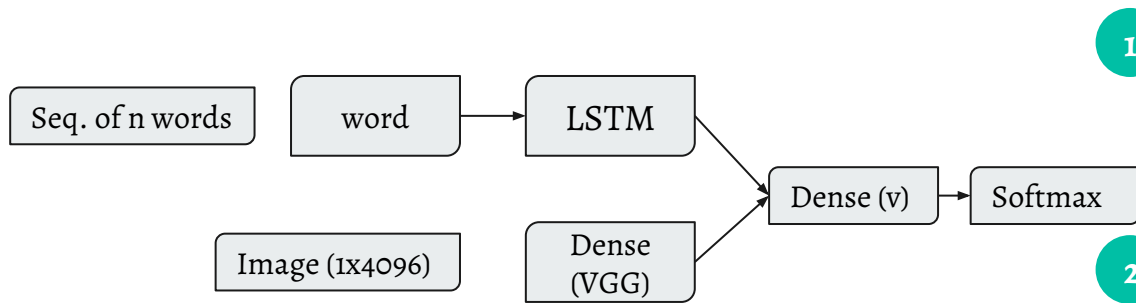
2

Conditioning by injecting the image means injecting the image into the same LSTM that processes the words.

3

In one class of architectures, image features are directly incorporated into the LSTM during the sequence encoding process.

1.6.2 Fitting the Enc-Dec LSTM model on both representations



Architecture 2: Merge

'Dense' means fully connected layer with bias and v is the vocab size

1

A Merge architecture keeps the encoding of linguistic and perceptual features separate, merging them in a later multimodal layer, at which point predictions are made.

2

In this type of model, the LSTM is functioning primarily as an encoder of sequences of word embeddings, with the visual features merged with the linguistic features in a later, multimodal layer.

3

This multimodal layer is the one that drives the generation process since the LSTM never sees the image and hence would not be able to direct the generation process.



1.7 Model Evaluation

1

BLEU (bilingual evaluation understudy) Quality is considered to be the correspondence between a machine's output and that of a human.

2

"The closer a machine translation is to a professional human translation, the better it is" – this is the central idea behind BLEU.

Thank you.

