Table of Contents

DL Assignment 2 - cnn_rnn_assignment_set_2

Group- 022

Sanka Mahesh Sai -- 2019ab04135

Snigdha Tarua -- 2019ab04171

Aravapalli Chandra Sekhar Gupta -- 2019ab04187

Question:

Image Captioning : Image Captioning is the process of generating textual description of an image. It uses both Natural Language Processing and Computer Vision to generate the captions. The dataset will be in the form [image → captions]. The dataset consists of input images and their corresponding output captions.

Encoder

The Convolutional Neural Network(CNN) can be thought of as an encoder. The input image is given to CNN to extract the features. The last hidden state of the CNN is connected to the Decoder.

Decoder

The Decoder is a Recurrent Neural Network(RNN) which does language modelling up to the word level. The first time step receives the encoded output from the encoder and also the vector.

1. Import Libraries/Dataset

a. Import the required libraries

b. Check the GPU available (recommended- use free GPU provided by Google Colab).

In [ ]:

```
#For SqeezeNet
#!pip install git+https://github.com/rcmalli/keras-squeezenet.git
```

In [ ]:

```
# checking for the gpu available
!nvidia-smi
```

In [1]:

```
from keras_squeezenet import SqueezeNet
```

```
Using TensorFlow backend.
```

In [2]:

```
import pickle
import matplotlib.pyplot as plt
import tensorflow as tf
import keras


import sys, time, os, warnings
import numpy as np
import pandas as pd
from collections import Counter
warnings.filterwarnings("ignore")
print("python {}".format(sys.version))
#print("keras version {}".format(keras.__version__)); del keras
print("tensorflow version {}".format(tf.__version__))
#config = tf.ConfigProto()
#config.gpu_options.per_process_gpu_memory_fraction = 0.95
#config.gpu_options.visible_device_list = "0"
#set_session(tf.Session(config=config))


def set_seed(sd=123):
    from numpy.random import seed
    from tensorflow import set_random_seed
    import random as rn
    ## numpy random seed
    seed(sd)
    ## core python's random number
    rn.seed(sd)
    ## tensor flow's random number
    set_random_seed(sd)
```

```
python 3.6.6 |Anaconda, Inc.| (default, Oct  9 2018, 12:34:16)
[GCC 7.3.0]
tensorflow version 1.13.1
```

2. . Data Visualization and augmentation

Reading the pickle and image files :

a. Plot at least two samples and their captions (use matplotlib/seaborn/any other library).

b. Bring the train and test data in the required format.

In [3]:

```
#Reading caption pickle file
captions = pickle.load(open("../input/image-caption/set_2.pkl","rb"))
captions[0]
```

Out[3]:

```
'3192266178_f9bf5d3dba.jpg#3\tA man in blue with a black hat with a do
g leap at him in a park-like set .'
```

In [4]:

```
# number of available captions
len(captions)
```

Out[4]:

```
25000
```

In [ ]:

```
#!unzip Image_captioning_Dataset.zip
```

In [5]:

```
IMAGES_PATH = "../input/flicker-images/Flicker8k_Dataset"

## The location of the Flickr8K_ photos
dir_Flickr_jpg = "../input/flicker-images/Flicker8k_Dataset/"
## The location of the caption file
dir_Flickr_text = captions

jpgs = os.listdir(dir_Flickr_jpg)
print("The number of jpg flies in Flicker8k: {}".format(len(jpgs)))
```

```
The number of jpg flies in Flicker8k: 8091
```

In [6]:

```
#convert the files to a proper structure for training
datatxt = []
for line in captions:
    col = line.split('\t')
    if len(col) == 1:
        continue
    w = col[0].split("#")
    datatxt.append(w + [col[1].lower()])

df_txt = pd.DataFrame(datatxt,columns=["filename","index","caption"])


uni_filenames = np.unique(df_txt.filename.values)
print("The number of unique file names : {}".format(len(uni_filenames)))
print("The distribution of the number of captions for each image:")
Counter(Counter(df_txt.filename.values).values())
```

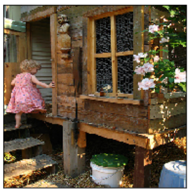
```
The number of unique file names : 8021
The distribution of the number of captions for each image:
```

Out[6]:

```
Counter({2: 1710, 3: 2832, 4: 2211, 5: 743, 1: 525})
```

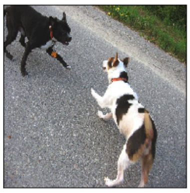
In [7]:

```
#Plotting of some sample images
from keras.preprocessing.image import load_img, img_to_array

npic = 5
npix = 224
target_size = (npix,npix,3)

count = 1
fig = plt.figure(figsize=(10,20))
for jpgfnm in uni_filenames[:npic]:
    filename = dir_Flickr_jpg + '/' + jpgfnm
    captions = list(df_txt["caption"].loc[df_txt["filename"]==jpgfnm].values)
    image_load = load_img(filename, target_size=target_size)

    ax = fig.add_subplot(npic,2,count,xticks=[],yticks=[])
    ax.imshow(image_load)
    count += 1

    ax = fig.add_subplot(npic,2,count)
    plt.axis('off')
    ax.plot()
    ax.set_xlim(0,1)
    ax.set_ylim(0,len(captions))
    for i, caption in enumerate(captions):
        ax.text(0,i,caption,fontsize=20)
    count += 1
plt.show()
```



a little girl climb into a wooden playhouse .

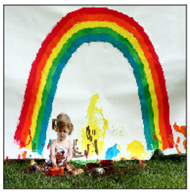
a little girl in a pink dress go into a wooden cabin .



two dog of different breed look at each other on a road .

a black dog and a white dog with brown spot be stare at each other in a street .

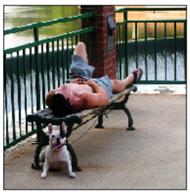
two dog on pavement move toward each other .



a small girl in the grass play with fingerpaint in front of a white canvas with a rainbow on it .

young girl with pigtail paint outside in the grass .

a little girl be sit in front of a large painted rainbow .

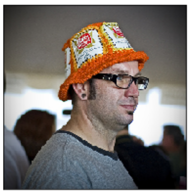


a shirtless man lie on a park bench with his dog .

man lay on bench hold leash of dog sit on ground

a man lay on a bench while his dog sit by him .

a man lay on a bench to which a white dog be also tie .



a man in an orange hat star at something .

a man with glasses be wear a beer can crochet hat .

a man with pierced ear be wear glasses and an orange hat .

In [8]:

```
#creating word-count
def df_word(df_txt):
    vocabulary = []
    for txt in df_txt.caption.values:
        vocabulary.extend(txt.split())
    print('Vocabulary Size: %d' % len(set(vocabulary)))
    ct = Counter(vocabulary)
    #dfword = pd.DataFrame({"word":ct.keys(),"count":ct.values()})
    df = pd.DataFrame.from_dict(ct, orient='index').reset_index()
    df.columns = ['word', 'count']
    #dfword = dfword.sort_index("count",ascending=False)
    #dfword = dfword.reset_index()[["word","count"]]
    return(df)

dfword = df_word(df_txt)
dfword.head(3)
```

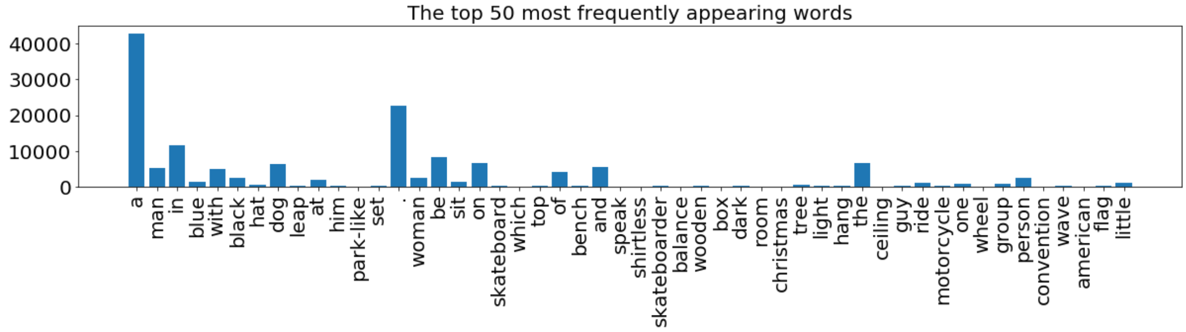
```
Vocabulary Size: 5588
```

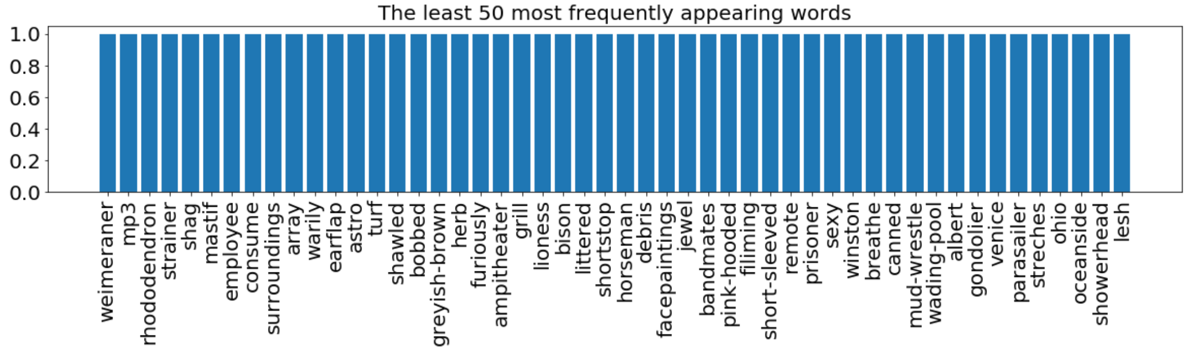
Out[8]:

	word	count
0	a	42947
1	man	5185
2	in	11620

In [9]:

```
#EDA OF THE CAPTION DATA
topn = 50

def plthist(dfsub, title="The top 50 most frequently appearing words"):
    plt.figure(figsize=(20,3))
    plt.bar(dfsub.index,dfsub["count"])
    plt.yticks(fontsize=20)
    plt.xticks(dfsub.index,dfsub["word"],rotation=90,fontsize=20)
    plt.title(title,fontsize=20)
    plt.show()

plthist(dfword.iloc[:topn,:],
        title="The top 50 most frequently appearing words")
plthist(dfword.iloc[-topn:,:],
        title="The least 50 most frequently appearing words")
```





In [10]:

```
#pre-processing of texts
import string
text_original = "I ate 1000 apples and a banana. I have python v2.7. It's 2:30 pm. C

print(text_original)
print("\nRemove punctuations..")
def remove_punctuation(text_original):
    text_no_punctuation = text_original.translate((str.maketrans('','',string.punctu
    return(text_no_punctuation)
text_no_punctuation = remove_punctuation(text_original)
print(text_no_punctuation)


print("\nRemove a single character word..")
def remove_single_character(text):
    text_len_more_than1 = ""
    for word in text.split():
        if len(word) > 1:
            text_len_more_than1 += " " + word
    return(text_len_more_than1)
text_len_more_than1 = remove_single_character(text_no_punctuation)
print(text_len_more_than1)

print("\nRemove words with numeric values..")
def remove_numeric(text,printTF=False):
    text_no_numeric = ""
    for word in text.split():
        isalpha = word.isalpha()
        if printTF:
            print("    {:10} : {:}".format(word,isalpha))
        if isalpha:
            text_no_numeric += " " + word
    return(text_no_numeric)
text_no_numeric = remove_numeric(text_len_more_than1,printTF=True)
print(text_no_numeric)
```

```
I ate 1000 apples and a banana. I have python v2.7. It's 2:30 pm. Coul
d you buy me iphone7?

Remove punctuations..
I ate 1000 apples and a banana I have python v27 Its 230 pm Could you
buy me iphone7

Remove a single character word..
 ate 1000 apples and banana have python v27 Its 230 pm Could you buy m
e iphone7

Remove words with numeric values..
    ate        : True
    1000       : False
    apples     : True
    and        : True
    banana     : True
    have       : True
    python     : True
    v27        : False
    Its        : True
    230        : False
    pm         : True
    Could      : True
    you        : True
    buy        : True
    me         : True
    iphone7    : False
 ate apples and banana have python Its pm Could you buy me
```

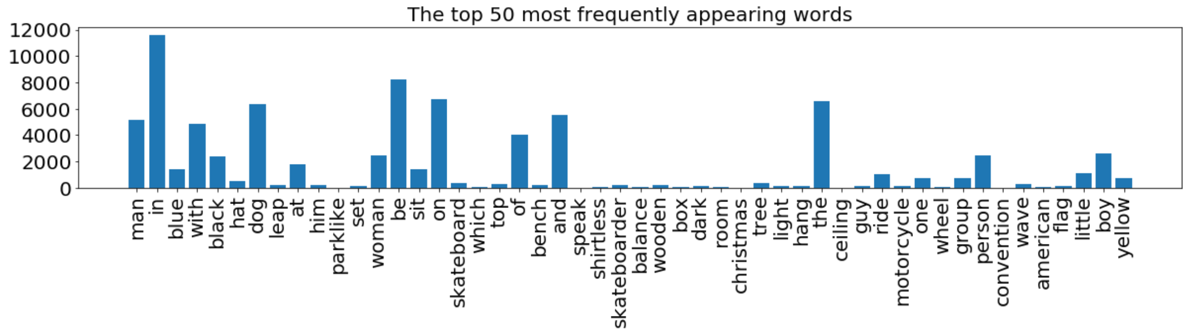
In [11]:

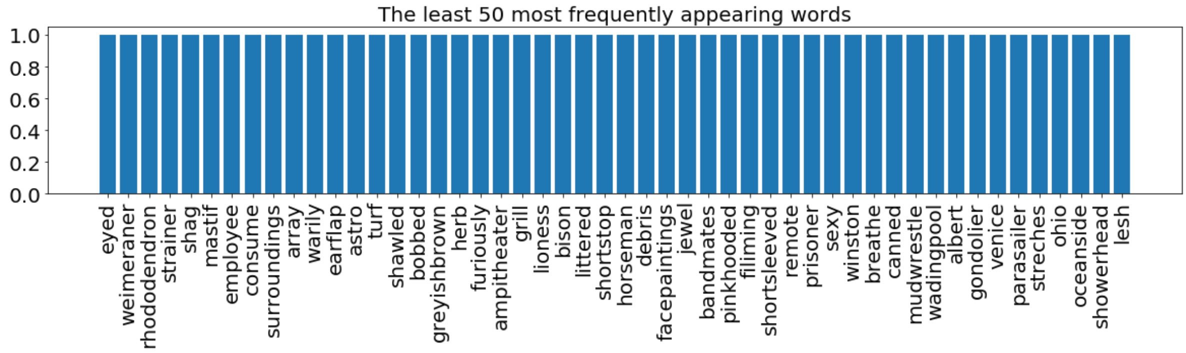
```
def text_clean(text_original):
    text = remove_punctuation(text_original)
    text = remove_single_character(text)
    text = remove_numeric(text)
    return(text)


for i, caption in enumerate(df_txt.caption.values):
    newcaption = text_clean(caption)
    df_txt["caption"].iloc[i] = newcaption
```

In [12]:

```
#plotting most/least frequent words in captions
dfword = df_word(df_txt)
plthist(dfword.iloc[:topn,:],
        title="The top 50 most frequently appearing words")
plthist(dfword.iloc[-topn:,:],
        title="The least 50 most frequently appearing words")
```

```
Vocabulary Size: 5514
```





In [14]:

```
#Adding start/end token
from copy import copy
def add_start_end_seq_token(captions):
    caps = []
    for txt in captions:
        txt = 'startseq ' + txt + ' endseq'
        caps.append(txt)
    return(caps)
df_txt0 = copy(df_txt)
df_txt0["caption"] = add_start_end_seq_token(df_txt["caption"])
```

In [15]:

```
df_txt0.head(5)
```

Out[15]:

	filename	index	caption
0	3192266178_f9bf5d3dba.jpg	3	startseq man in blue with black hat with dog ...
1	532457586_bddfc5251d.jpg	4	startseq woman with hat be sit on skateboard ...
2	3218889785_86cb64014f.jpg	2	startseq skateboarder be balance on wooden bo...
3	2217728745_92b6779016.jpg	4	startseq christmas tree light hang on the cei...
4	2616508003_fa5ca5780d.jpg	0	startseq guy ride motorcycle on one wheel endseq

3. Model Building

a. Use Pretrained Squeezenet model trained on ImageNet dataset (available publicly on google) for image feature extraction.

b. Create 3 layered GRU layer model and other relevant layers for image caption generation.

d. Add one layer of dropout at the appropriate position and give reasons.

e. Choose the appropriate activation function for all the layers.

f. Print the model summary.

In [16]:

```
#Using pretrained squeezenet model
#using keras and tensorflow v1 due to unavailability of  pretrained sqeezenet is in
modelvgg = SqueezeNet(include_top=True, weights='imagenet')
modelvgg.summary()
```

```
WARNING:tensorflow:From /opt/conda/lib/python3.6/site-packages/tensorf
low/python/framework/op_def_library.py:263: colocate_with (from tensor
flow.python.framework.ops) is deprecated and will be removed in a futu
re version.
Instructions for updating:
Colocations handled automatically by placer.
WARNING:tensorflow:From /opt/conda/lib/python3.6/site-packages/keras/b
ackend/tensorflow_backend.py:3445: calling dropout (from tensorflow.py
thon.ops.nn_ops) with keep_prob is deprecated and will be removed in a
future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate
= 1 - keep_prob`.
______________________________________________________________________
____________________________
Layer (type)                    Output Shape         Param #     Conne
cted to
======================================================================
============================
input_1 (InputLayer)            (None, 227, 227, 3)  0
______________________________________________________________________
____________________________
conv1 (Conv2D)                  (None, 113, 113, 64) 1792        input
_1[0][0]
______________________________________________________________________
____________________________
relu_conv1 (Activation)         (None, 113, 113, 64) 0           conv1
[0][0]
______________________________________________________________________
____________________________
pool1 (MaxPooling2D)            (None, 56, 56, 64)   0           relu_
conv1[0][0]
______________________________________________________________________
____________________________
fire2/squeeze1x1 (Conv2D)       (None, 56, 56, 16)   1040        pool1
[0][0]
______________________________________________________________________
____________________________
fire2/relu_squeeze1x1 (Activati (None, 56, 56, 16)   0           fire
2/squeeze1x1[0][0]
______________________________________________________________________
____________________________
fire2/expand1x1 (Conv2D)        (None, 56, 56, 64)   1088        fire
2/relu_squeeze1x1[0][0]
______________________________________________________________________
____________________________
fire2/expand3x3 (Conv2D)        (None, 56, 56, 64)   9280        fire
2/relu_squeeze1x1[0][0]
______________________________________________________________________
____________________________
fire2/relu_expand1x1 (Activatio (None, 56, 56, 64)   0           fire
2/expand1x1[0][0]
______________________________________________________________________
____________________________
```

```
fire2/relu_expand3x3 (Activatio (None, 56, 56, 64)   0           fire
2/expand3x3[0][0]
______________________________________________________________________
____________________________
fire2/concat (Concatenate)      (None, 56, 56, 128)  0           fire
2/relu_expand1x1[0][0]
                                                                 fire
2/relu_expand3x3[0][0]
______________________________________________________________________
____________________________
fire3/squeeze1x1 (Conv2D)       (None, 56, 56, 16)   2064        fire
2/concat[0][0]
______________________________________________________________________
____________________________
fire3/relu_squeeze1x1 (Activati (None, 56, 56, 16)   0           fire
3/squeeze1x1[0][0]
______________________________________________________________________
____________________________
fire3/expand1x1 (Conv2D)        (None, 56, 56, 64)   1088        fire
3/relu_squeeze1x1[0][0]
______________________________________________________________________
____________________________
fire3/expand3x3 (Conv2D)        (None, 56, 56, 64)   9280        fire
3/relu_squeeze1x1[0][0]
______________________________________________________________________
____________________________
fire3/relu_expand1x1 (Activatio (None, 56, 56, 64)   0           fire
3/expand1x1[0][0]
______________________________________________________________________
____________________________
fire3/relu_expand3x3 (Activatio (None, 56, 56, 64)   0           fire
3/expand3x3[0][0]
______________________________________________________________________
____________________________
fire3/concat (Concatenate)      (None, 56, 56, 128)  0           fire
3/relu_expand1x1[0][0]
                                                                 fire
3/relu_expand3x3[0][0]
______________________________________________________________________
____________________________
pool3 (MaxPooling2D)            (None, 27, 27, 128)  0           fire
3/concat[0][0]
______________________________________________________________________
____________________________
fire4/squeeze1x1 (Conv2D)       (None, 27, 27, 32)   4128        pool3
[0][0]
______________________________________________________________________
____________________________
fire4/relu_squeeze1x1 (Activati (None, 27, 27, 32)   0           fire
4/squeeze1x1[0][0]
______________________________________________________________________
____________________________
fire4/expand1x1 (Conv2D)        (None, 27, 27, 128)  4224        fire
4/relu_squeeze1x1[0][0]
______________________________________________________________________
____________________________
fire4/expand3x3 (Conv2D)        (None, 27, 27, 128)  36992       fire
4/relu_squeeze1x1[0][0]
______________________________________________________________________
____________________________
fire4/relu_expand1x1 (Activatio (None, 27, 27, 128)  0           fire
4/expand1x1[0][0]
______________________________________________________________________
____________________________
fire4/relu_expand3x3 (Activatio (None, 27, 27, 128)  0           fire
4/expand3x3[0][0]
______________________________________________________________________
____________________________
fire4/concat (Concatenate)      (None, 27, 27, 256)  0           fire
4/relu_expand1x1[0][0]
                                                                 fire
4/relu_expand3x3[0][0]
______________________________________________________________________
____________________________
fire5/squeeze1x1 (Conv2D)       (None, 27, 27, 32)   8224        fire
4/concat[0][0]
______________________________________________________________________
____________________________
fire5/relu_squeeze1x1 (Activati (None, 27, 27, 32)   0           fire
5/squeeze1x1[0][0]
______________________________________________________________________
____________________________
fire5/expand1x1 (Conv2D)        (None, 27, 27, 128)  4224        fire
5/relu_squeeze1x1[0][0]
______________________________________________________________________
____________________________
fire5/expand3x3 (Conv2D)        (None, 27, 27, 128)  36992       fire
5/relu_squeeze1x1[0][0]
______________________________________________________________________
____________________________
fire5/relu_expand1x1 (Activatio (None, 27, 27, 128)  0           fire
5/expand1x1[0][0]
______________________________________________________________________
____________________________
fire5/relu_expand3x3 (Activatio (None, 27, 27, 128)  0           fire
5/expand3x3[0][0]
______________________________________________________________________
____________________________
fire5/concat (Concatenate)      (None, 27, 27, 256)  0           fire
5/relu_expand1x1[0][0]
                                                                 fire
5/relu_expand3x3[0][0]
______________________________________________________________________
____________________________
pool5 (MaxPooling2D)            (None, 13, 13, 256)  0           fire
5/concat[0][0]
______________________________________________________________________
____________________________
fire6/squeeze1x1 (Conv2D)       (None, 13, 13, 48)   12336       pool5
[0][0]
______________________________________________________________________
____________________________
fire6/relu_squeeze1x1 (Activati (None, 13, 13, 48)   0           fire
6/squeeze1x1[0][0]
______________________________________________________________________
____________________________
fire6/expand1x1 (Conv2D)        (None, 13, 13, 192)  9408        fire
6/relu_squeeze1x1[0][0]
______________________________________________________________________
____________________________
fire6/expand3x3 (Conv2D)        (None, 13, 13, 192)  83136       fire
6/relu_squeeze1x1[0][0]
```

```
______________________________________________________________________
____________________________
fire6/relu_expand1x1 (Activatio (None, 13, 13, 192)  0           fire
6/expand1x1[0][0]

______________________________________________________________________
____________________________
fire6/relu_expand3x3 (Activatio (None, 13, 13, 192)  0           fire
6/expand3x3[0][0]

______________________________________________________________________
____________________________
fire6/concat (Concatenate)      (None, 13, 13, 384)  0           fire
6/relu_expand1x1[0][0]
                                                                 fire
6/relu_expand3x3[0][0]

______________________________________________________________________
____________________________
fire7/squeeze1x1 (Conv2D)       (None, 13, 13, 48)   18480       fire
6/concat[0][0]

______________________________________________________________________
____________________________
fire7/relu_squeeze1x1 (Activati (None, 13, 13, 48)   0           fire
7/squeeze1x1[0][0]

______________________________________________________________________
____________________________
fire7/expand1x1 (Conv2D)        (None, 13, 13, 192)  9408        fire
7/relu_squeeze1x1[0][0]

______________________________________________________________________
____________________________
fire7/expand3x3 (Conv2D)        (None, 13, 13, 192)  83136       fire
7/relu_squeeze1x1[0][0]

______________________________________________________________________
____________________________
fire7/relu_expand1x1 (Activatio (None, 13, 13, 192)  0           fire
7/expand1x1[0][0]

______________________________________________________________________
____________________________
fire7/relu_expand3x3 (Activatio (None, 13, 13, 192)  0           fire
7/expand3x3[0][0]

______________________________________________________________________
____________________________
fire7/concat (Concatenate)      (None, 13, 13, 384)  0           fire
7/relu_expand1x1[0][0]
                                                                 fire
7/relu_expand3x3[0][0]

______________________________________________________________________
____________________________
fire8/squeeze1x1 (Conv2D)       (None, 13, 13, 64)   24640       fire
7/concat[0][0]

______________________________________________________________________
____________________________
fire8/relu_squeeze1x1 (Activati (None, 13, 13, 64)   0           fire
8/squeeze1x1[0][0]

______________________________________________________________________
____________________________
fire8/expand1x1 (Conv2D)        (None, 13, 13, 256)  16640       fire
8/relu_squeeze1x1[0][0]

______________________________________________________________________
____________________________
fire8/expand3x3 (Conv2D)        (None, 13, 13, 256)  147712      fire
8/relu_squeeze1x1[0][0]

______________________________________________________________________
```

```
____________________________
fire8/relu_expand1x1 (Activatio (None, 13, 13, 256)  0           fire
8/expand1x1[0][0]

_______________________________________________________________________
____________________________
fire8/relu_expand3x3 (Activatio (None, 13, 13, 256)  0           fire
8/expand3x3[0][0]

_______________________________________________________________________
____________________________
fire8/concat (Concatenate)      (None, 13, 13, 512)  0           fire
8/relu_expand1x1[0][0]
                                                                 fire
8/relu_expand3x3[0][0]

_______________________________________________________________________
____________________________
fire9/squeeze1x1 (Conv2D)       (None, 13, 13, 64)   32832       fire
8/concat[0][0]

_______________________________________________________________________
____________________________
fire9/relu_squeeze1x1 (Activati (None, 13, 13, 64)   0           fire
9/squeeze1x1[0][0]

_______________________________________________________________________
____________________________
fire9/expand1x1 (Conv2D)        (None, 13, 13, 256)  16640       fire
9/relu_squeeze1x1[0][0]

_______________________________________________________________________
____________________________
fire9/expand3x3 (Conv2D)        (None, 13, 13, 256)  147712      fire
9/relu_squeeze1x1[0][0]

_______________________________________________________________________
____________________________
fire9/relu_expand1x1 (Activatio (None, 13, 13, 256)  0           fire
9/expand1x1[0][0]

_______________________________________________________________________
____________________________
fire9/relu_expand3x3 (Activatio (None, 13, 13, 256)  0           fire
9/expand3x3[0][0]

_______________________________________________________________________
____________________________
fire9/concat (Concatenate)      (None, 13, 13, 512)  0           fire
9/relu_expand1x1[0][0]
                                                                 fire
9/relu_expand3x3[0][0]

_______________________________________________________________________
____________________________
drop9 (Dropout)                 (None, 13, 13, 512)  0           fire
9/concat[0][0]

_______________________________________________________________________
____________________________
conv10 (Conv2D)                 (None, 13, 13, 1000) 513000      drop9
[0][0]

_______________________________________________________________________
____________________________
relu_conv10 (Activation)        (None, 13, 13, 1000) 0           conv1
0[0][0]

_______________________________________________________________________
____________________________
global_average_pooling2d_1 (Glo (None, 1000)         0           relu_
conv10[0][0]

_______________________________________________________________________
____________________________
```

```
loss (Activation)               (None, 1000)         0           globa
l_average_pooling2d_1[0][0]
======================================================================
============================
Total params: 1,235,496
Trainable params: 1,235,496
Non-trainable params: 0
______________________________________________________________________
____________________________
```

In [23]:

```
#Using VGG Pretrained model for comaprison
!wget 'https://github.com/fchollet/deep-learning-models/releases/download/v0.1/vgg16
from keras.applications import VGG16

modelvgg = VGG16(include_top=True,weights=None)
## load the locally saved weights
modelvgg.load_weights("vgg16_weights_tf_dim_ordering_tf_kernels.h5")
modelvgg.summary()

from keras import models
modelvgg.layers.pop()
modelvgg = models.Model(inputs=modelvgg.inputs, outputs=modelvgg.layers[-1].output)
## show the deep learning model
modelvgg.summary()
```

```
--2021-08-08 08:45:34--  https://github.com/fchollet/deep-learning-mod
els/releases/download/v0.1/vgg16_weights_tf_dim_ordering_tf_kernels.h5
(https://github.com/fchollet/deep-learning-models/releases/download/v
0.1/vgg16_weights_tf_dim_ordering_tf_kernels.h5)
Resolving github.com (github.com)... 140.82.114.3
Connecting to github.com (github.com)|140.82.114.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github-releases.githubusercontent.com/64878964/b0afb
ae8-5983-11e6-90f4-e3db656bd548?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz
-Credential=AKIAIWNJYAX4CSVEH53A%2F20210808%2Fus-east-1%2Fs3%2Faws4_re
quest&X-Amz-Date=20210808T084439Z&X-Amz-Expires=300&X-Amz-Signature=dd
e9469ab04fab5da8a45ebdd85c5a37951e7b2cb8c2dbfd502da11030753cd6&X-Amz-S
ignedHeaders=host&actor_id=0&key_id=0&repo_id=64878964&response-conten
t-disposition=attachment%3B%20filename%3Dvgg16_weights_tf_dim_ordering
_tf_kernels.h5&response-content-type=application%2Foctet-stream (http
s://github-releases.githubusercontent.com/64878964/b0afbae8-5983-11e6-
90f4-e3db656bd548?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AK
IAIWNJYAX4CSVEH53A%2F20210808%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Da
te=20210808T084439Z&X-Amz-Expires=300&X-Amz-Signature=dde9469ab04fab5d
a8a45ebdd85c5a37951e7b2cb8c2dbfd502da11030753cd6&X-Amz-SignedHeaders=h
ost&actor_id=0&key_id=0&repo_id=64878964&response-content-disposition=
attachment%3B%20filename%3Dvgg16_weights_tf_dim_ordering_tf_kernels.h5
&response-content-type=application%2Foctet-stream) [following]
--2021-08-08 08:45:34--  https://github-releases.githubusercontent.co
m/64878964/b0afbae8-5983-11e6-90f4-e3db656bd548?X-Amz-Algorithm=AWS4-H
MAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20210808%2Fus-east-
1%2Fs3%2Faws4_request&X-Amz-Date=20210808T084439Z&X-Amz-Expires=300&X-
Amz-Signature=dde9469ab04fab5da8a45ebdd85c5a37951e7b2cb8c2dbfd502da110
30753cd6&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=64878964
&response-content-disposition=attachment%3B%20filename%3Dvgg16_weights
_tf_dim_ordering_tf_kernels.h5&response-content-type=application%2Foct
et-stream (https://github-releases.githubusercontent.com/64878964/b0af
bae8-5983-11e6-90f4-e3db656bd548?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Am
z-Credential=AKIAIWNJYAX4CSVEH53A%2F20210808%2Fus-east-1%2Fs3%2Faws4_r
equest&X-Amz-Date=20210808T084439Z&X-Amz-Expires=300&X-Amz-Signature=d
de9469ab04fab5da8a45ebdd85c5a37951e7b2cb8c2dbfd502da11030753cd6&X-Amz-
SignedHeaders=host&actor_id=0&key_id=0&repo_id=64878964&response-conte
nt-disposition=attachment%3B%20filename%3Dvgg16_weights_tf_dim_orderin
g_tf_kernels.h5&response-content-type=application%2Foctet-stream)
Resolving github-releases.githubusercontent.com (github-releases.githu
busercontent.com)... 185.199.109.154, 185.199.108.154, 185.199.110.15
4, ...
Connecting to github-releases.githubusercontent.com (github-releases.g
ithubusercontent.com)|185.199.109.154|:443... connected.
```

```
HTTP request sent, awaiting response... 200 OK
Length: 553467096 (528M) [application/octet-stream]
Saving to: ‘vgg16_weights_tf_dim_ordering_tf_kernels.h5.2’

vgg16_weights_tf_di 100%[===================>] 527.83M   238MB/s    in
2.2s

2021-08-08 08:45:37 (238 MB/s) - ‘vgg16_weights_tf_dim_ordering_tf_ker
nels.h5.2’ saved [553467096/553467096]


_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input_5 (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         
_________________________________________________________________
flatten (Flatten)            (None, 25088)             0         
_________________________________________________________________
fc1 (Dense)                  (None, 4096)              102764544 
_________________________________________________________________
fc2 (Dense)                  (None, 4096)              16781312  
_________________________________________________________________
predictions (Dense)          (None, 1000)              4097000   
=================================================================
Total params: 138,357,544
Trainable params: 138,357,544
```

```
Non-trainable params: 0
_________________________________________________________________
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input_5 (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         
_________________________________________________________________
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
_________________________________________________________________
```

In [25]:

```
#creation of image features using pretrained model
from keras.preprocessing.image import load_img, img_to_array
from keras.applications.vgg16 import preprocess_input
from collections import OrderedDict

images = OrderedDict()
npix = 227
target_size = (npix,npix,3)
data = np.zeros((len(jpgs), npix,npix,3))
for i,name in enumerate(jpgs):
    # load an image from file
    filename = dir_Flickr_jpg + '/' + name
    image = load_img(filename, target_size=target_size)
    # convert the image pixels to a numpy array
    image = img_to_array(image)
    nimage = preprocess_input(image)

    y_pred = modelvgg.predict(nimage.reshape( (1,) + nimage.shape[:3]))
    images[name] = y_pred.flatten()
```

In [28]:

```
dimages, keepindex = [],[]
df_txt0 = df_txt0.loc[df_txt0["index"].values == "0",: ]
for i, fnm in enumerate(df_txt0.filename):
    if fnm in images.keys():
        dimages.append(images[fnm])
        keepindex.append(i)

fnames = df_txt0["filename"].iloc[keepindex].values
dcaptions = df_txt0["caption"].iloc[keepindex].values
dimages = np.array(dimages)
```

In [29]:

```
#using keras function for text preprocessing
from keras.preprocessing.text import Tokenizer

## the maximum number of words in dictionary
nb_words = 8000
tokenizer = Tokenizer(nb_words=nb_words)
tokenizer.fit_on_texts(dcaptions)
vocab_size = len(tokenizer.word_index) + 1
print("vocabulary size : {}".format(vocab_size))
dtexts = tokenizer.texts_to_sequences(dcaptions)
print(dtexts[:5])
```

```
vocabulary size : 2762
[[1, 127, 42, 182, 7, 87, 513, 2], [1, 16, 6, 15, 5, 4, 19, 54, 147, 1
30, 2], [1, 29, 51, 3, 186, 6, 27, 212, 51, 3, 13, 2], [1, 5, 590, 7,
25, 138, 3, 8, 43, 64, 22, 82, 5, 47, 7, 2], [1, 12, 42, 25, 75, 54, 2
7, 1483, 79, 17, 187, 2]]
```

In [30]:

```
#Test/train/validation split
prop_test, prop_val = 0.2, 0.2

N = len(dtexts)
Ntest, Nval = int(N*prop_test), int(N*prop_val)

def split_test_val_train(dtexts,Ntest,Nval):
    return(dtexts[:Ntest],
           dtexts[Ntest:Ntest+Nval],
           dtexts[Ntest+Nval:])

dt_test,  dt_val, dt_train   = split_test_val_train(dtexts,Ntest,Nval)
di_test,  di_val, di_train   = split_test_val_train(dimages,Ntest,Nval)
fnm_test,fnm_val, fnm_train  = split_test_val_train(fnames,Ntest,Nval)
```

In [31]:

```
maxlen = np.max([len(text) for text in dtexts])
```

In [32]:

```
from keras.preprocessing.sequence import pad_sequences
from keras.utils import to_categorical

def preprocessing(dtexts,dimages):
    N = len(dtexts)
    print("# captions/images = {}".format(N))

    assert(N==len(dimages))
    Xtext, Ximage, ytext = [],[],[]
    for text,image in zip(dtexts,dimages):

        for i in range(1,len(text)):
            in_text, out_text = text[:i], text[i]
            in_text = pad_sequences([in_text],maxlen=maxlen).flatten()
            out_text = to_categorical(out_text,num_classes = vocab_size)

            Xtext.append(in_text)
            Ximage.append(image)
            ytext.append(out_text)

    Xtext  = np.array(Xtext)
    Ximage = np.array(Ximage)
    ytext  = np.array(ytext)
    print(" {} {} {}".format(Xtext.shape,Ximage.shape,ytext.shape))
    return(Xtext,Ximage,ytext)


Xtext_train, Ximage_train, ytext_train = preprocessing(dt_train,di_train)
Xtext_val,   Ximage_val,   ytext_val   = preprocessing(dt_val,di_val)
#pre-processing is not necessary for testing data
#Xtext_test,  Ximage_test,  ytext_test  = preprocessing(dt_test,di_test)
```

```
# captions/images = 3013
 (30503, 30) (30503, 4096) (30503, 2762)
# captions/images = 1004
 (10191, 30) (10191, 4096) (10191, 2762)
```

4. Model Compilation

a. Compile the model with the appropriate loss function.

b. Use an appropriate optimizer. Give reasons for the choice of learning rate and its value.

In [33]:

```
#Defining model : Sequential / Encoder / Decoder layers
rom keras import layers
rom keras import models

rint(vocab_size)
## image feature
im_embedding = 64
nput_image = layers.Input(shape=(Ximage_train.shape[1],))
eshape_input = layers.Reshape((-1,Ximage_train.shape[1]))(input_image)
image = layers.GRU(256,activation='relu',return_sequences=True, kernel_regularizer=t

## sequence model
nput_txt = layers.Input(shape=(maxlen,))
txt = layers.Embedding(vocab_size,dim_embedding, mask_zero=True)(input_txt)
txt_drop = layers.Dropout(0.5)(ftxt)
txt = layers.GRU(256,return_sequences=True, kernel_regularizer=tf.keras.regularizers

## combined model for decoder
ecoder = layers.add([ftxt,fimage])
ecoder_gru = layers.GRU(256, activation='relu',kernel_regularizer=tf.keras.regulariz
#decoder = layers.SimpleRNN(256)(decoder_gru)
utput = layers.Dense(vocab_size,activation='softmax')(decoder_gru)
odel = models.Model(inputs=[input_image, input_txt],outputs=output)

odel.compile(loss='categorical_crossentropy', optimizer='adam(learning_rate=0.001)',

rint(model.summary())
```

```
2762
_______________________________________________________________________
_____________________________
Layer (type)                    Output Shape         Param #     Conne
cted to
=======================================================================
=============================
input_7 (InputLayer)            (None, 30)           0
_______________________________________________________________________
_____________________________
embedding_1 (Embedding)         (None, 30, 64)       176768      input
_7[0][0]
_______________________________________________________________________
_____________________________
input_6 (InputLayer)            (None, 4096)         0
_______________________________________________________________________
_____________________________
dropout_1 (Dropout)             (None, 30, 64)       0           embed
ding_1[0][0]
_______________________________________________________________________
_____________________________
reshape_1 (Reshape)             (None, 1, 4096)      0           input
_6[0][0]
_______________________________________________________________________
_____________________________
gru_2 (GRU)                     (None, 30, 256)      246528      dropo
ut_1[0][0]
_______________________________________________________________________
_____________________________
gru_1 (GRU)                     (None, 1, 256)       3343104     resha
pe_1[0][0]
```

```
______________________________________________________________________
____________________________
add_1 (Add)                     (None, 30, 256)      0           gru_2
[0][0]
                                                                 gru_1
[0][0]
______________________________________________________________________
____________________________
gru_3 (GRU)                     (None, 256)          393984      add_1
[0][0]
______________________________________________________________________
____________________________
dense_1 (Dense)                 (None, 2762)         709834      gru_3
[0][0]
======================================================================
============================
Total params: 4,870,218
Trainable params: 4,870,218
Non-trainable params: 0
______________________________________________________________________
____________________________
None
```

```
## Reasons:

* We have used adam after testing on rmsprop and madam optimizers as adam is
converging faster.

* We have used a learning rate of 1e-3 as we have to find balence between low and
high learning rate so that it finds optimal point in our loss function in less
epochs due to resource constraint.

* Dropout is used as a way to control overfitting -- we have placed at the second
gru layer.

* We have placed it at the second layer of gru so that model don't overfit in
learning the abstract parts of the sentences.
```

5. Model Training (1 mark)

a. Train the model for an appropriate number of epochs. Print the train and validation loss for each epoch. Use the appropriate batch size.

b. Plot the loss and accuracy history graphs for both train and validation set. Print the total time taken for training

In [44]:

```
# fit model
start = time.time()
hist = model.fit([Ximage_train, Xtext_train], ytext_train,
                  epochs=50 , verbose=2,
                  batch_size=64,
                  validation_data=([Ximage_val, Xtext_val], ytext_val))
end = time.time()
print("TIME TOOK {:3.2f}MIN".format((end - start )/60))
```

```
Train on 30503 samples, validate on 10191 samples
Epoch 1/50
 - 47s - loss: 4.2293 - acc: 0.2177 - val_loss: 4.8388 - val_acc: 0.21
13
Epoch 2/50
 - 48s - loss: 4.0581 - acc: 0.2330 - val_loss: 4.8339 - val_acc: 0.21
96
Epoch 3/50
 - 49s - loss: 3.9054 - acc: 0.2379 - val_loss: 4.8691 - val_acc: 0.22
61
Epoch 4/50
 - 48s - loss: 3.8052 - acc: 0.2421 - val_loss: 4.8793 - val_acc: 0.22
15
Epoch 5/50
 - 49s - loss: 3.6870 - acc: 0.2474 - val_loss: 4.9356 - val_acc: 0.22
10
Epoch 6/50
 - 48s - loss: 3.5930 - acc: 0.2502 - val_loss: 5.0010 - val_acc: 0.22
43
Epoch 7/50
 - 47s - loss: 3.4925 - acc: 0.2575 - val_loss: 5.0471 - val_acc: 0.22
04
Epoch 8/50
 - 48s - loss: 3.4188 - acc: 0.2626 - val_loss: 5.0886 - val_acc: 0.21
94
Epoch 9/50
 - 48s - loss: 3.3749 - acc: 0.2646 - val_loss: 5.1514 - val_acc: 0.21
47
Epoch 10/50
 - 49s - loss: 3.2593 - acc: 0.2770 - val_loss: 5.2158 - val_acc: 0.21
58
Epoch 11/50
- 48s - loss: 3.1981 - acc: 0.2834 - val_loss: 5.3012 - val_acc: 0.21
29
Epoch 12/50
 - 47s - loss: 3.1441 - acc: 0.2912 - val_loss: 5.3821 - val_acc: 0.21
08
Epoch 13/50
 - 48s - loss: 3.0823 - acc: 0.2976 - val_loss: 5.3300 - val_acc: 0.21
08
Epoch 14/50
 - 49s - loss: 3.0169 - acc: 0.3088 - val_loss: 5.5094 - val_acc: 0.20
83
Epoch 15/50
 - 48s - loss: 2.9781 - acc: 0.3142 - val_loss: 5.5175 - val_acc: 0.20
99
Epoch 16/50
 - 48s - loss: 2.9346 - acc: 0.3235 - val_loss: 5.5498 - val_acc: 0.21
09
Epoch 17/50
```

```
 - 48s - loss: 2.9127 - acc: 0.3283 - val_loss: 5.5944 - val_acc: 0.21
53
Epoch 18/50
 - 47s - loss: 2.8474 - acc: 0.3332 - val_loss: 5.6314 - val_acc: 0.20
44
Epoch 19/50
 - 49s - loss: 2.8051 - acc: 0.3418 - val_loss: 5.6827 - val_acc: 0.20
68
Epoch 20/50
 - 49s - loss: 2.7886 - acc: 0.3442 - val_loss: 5.6575 - val_acc: 0.20
73
Epoch 21/50
 - 48s - loss: 2.7766 - acc: 0.3514 - val_loss: 5.6951 - val_acc: 0.20
76
Epoch 22/50
 - 48s - loss: 2.6718 - acc: 0.3669 - val_loss: 5.7144 - val_acc: 0.20
83
Epoch 25/50
 - 49s - loss: 2.6657 - acc: 0.3726 - val_loss: 5.8741 - val_acc: 0.20
54
Epoch 26/50
 - 48s - loss: 2.6356 - acc: 0.3778 - val_loss: 5.8233 - val_acc: 0.20
40
Epoch 27/50
 - 49s - loss: 2.5994 - acc: 0.3801 - val_loss: 5.8045 - val_acc: 0.20
68
Epoch 28/50
 - 48s - loss: 2.5958 - acc: 0.3809 - val_loss: 5.8989 - val_acc: 0.21
02
Epoch 29/50
 - 50s - loss: 2.5454 - acc: 0.3920 - val_loss: 5.9159 - val_acc: 0.20
09
Epoch 30/50
 - 48s - loss: 2.5465 - acc: 0.3965 - val_loss: 5.9732 - val_acc: 0.20
36
Epoch 31/50
 - 48s - loss: 2.5121 - acc: 0.3972 - val_loss: 5.9500 - val_acc: 0.20
35
Epoch 32/50
 - 49s - loss: 2.5299 - acc: 0.4020 - val_loss: 5.8994 - val_acc: 0.20
93
Epoch 33/50
 - 47s - loss: 2.4829 - acc: 0.4046 - val_loss: 6.0245 - val_acc: 0.20
09
Epoch 34/50
 - 49s - loss: 2.4719 - acc: 0.4074 - val_loss: 5.9966 - val_acc: 0.20
68
Epoch 35/50
 - 49s - loss: 2.4753 - acc: 0.4114 - val_loss: 6.0977 - val_acc: 0.20
31
Epoch 36/50
 - 47s - loss: 2.4363 - acc: 0.4144 - val_loss: 6.0567 - val_acc: 0.19
47
Epoch 37/50
 - 49s - loss: 2.4307 - acc: 0.4182 - val_loss: 6.1383 - val_acc: 0.20
22
Epoch 38/50
 - 48s - loss: 2.4102 - acc: 0.4224 - val_loss: 6.1616 - val_acc: 0.20
71
Epoch 39/50
 - 48s - loss: 2.4230 - acc: 0.4217 - val_loss: 6.0362 - val_acc: 0.20
```

```
40
Epoch 40/50
 - 49s - loss: 2.3781 - acc: 0.4248 - val_loss: 6.0502 - val_acc: 0.20
24
Epoch 41/50
 - 48s - loss: 2.3868 - acc: 0.4256 - val_loss: 6.2101 - val_acc: 0.20
33
Epoch 42/50
 - 48s - loss: 2.3892 - acc: 0.4279 - val_loss: 6.1835 - val_acc: 0.20
48
Epoch 43/50
 - 49s - loss: 2.3642 - acc: 0.4313 - val_loss: 6.1125 - val_acc: 0.21
16
Epoch 44/50
 - 48s - loss: 2.3722 - acc: 0.4352 - val_loss: 6.1510 - val_acc: 0.19
71
Epoch 45/50
 - 48s - loss: 2.3403 - acc: 0.4382 - val_loss: 6.2711 - val_acc: 0.19
36
Epoch 46/50
 - 49s - loss: 2.3345 - acc: 0.4358 - val_loss: 6.1685 - val_acc: 0.19
99
Epoch 47/50
 - 48s - loss: 2.3198 - acc: 0.4426 - val_loss: 6.2429 - val_acc: 0.20
24
Epoch 48/50
 - 48s - loss: 2.2982 - acc: 0.4481 - val_loss: 6.2349 - val_acc: 0.20
34
Epoch 49/50
 - 49s - loss: 2.3905 - acc: 0.4329 - val_loss: 6.3766 - val_acc: 0.19
82
Epoch 50/50
 - 49s - loss: 2.4173 - acc: 0.4310 - val_loss: 6.0707 - val_acc: 0.20
51
TIME TOOK 40.22MIN
```

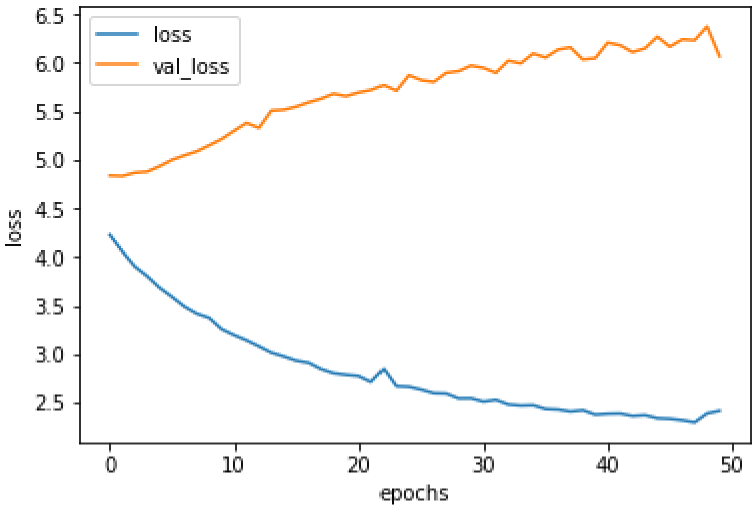
In [45]:

```
print(Ximage_train.shape,Xtext_train.shape,ytext_train.shape)
```

```
(30503, 4096) (30503, 30) (30503, 2762)
```

In [46]:

```
for label in ["loss","val_loss"]:
    plt.plot(hist.history[label],label=label)
plt.legend()
plt.xlabel("epochs")
plt.ylabel("loss")
plt.show()
```



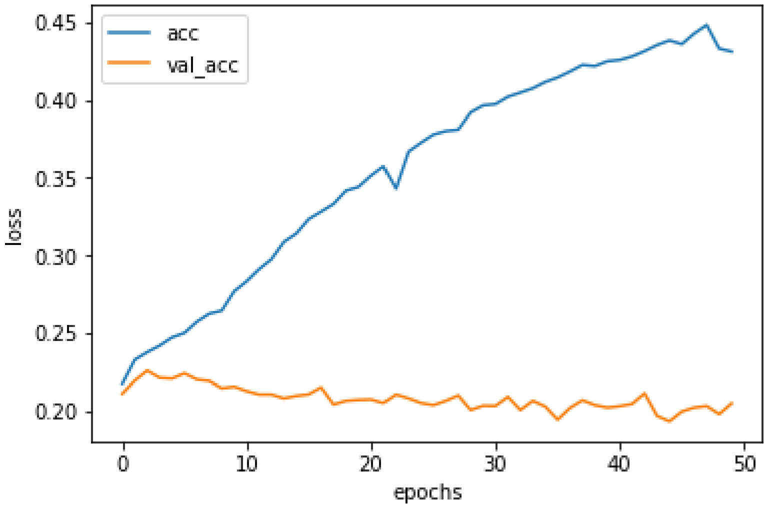
In [49]:

```
history_dict = hist.history
print(history_dict.keys())
```

```
dict_keys(['val_loss', 'val_acc', 'loss', 'acc'])
```

In [50]:

```
#  "Accuracy"
for label in ["acc","val_acc"]:
    plt.plot(hist.history[label],label=label)
plt.legend()
plt.xlabel("epochs")
plt.ylabel("loss")
plt.show()
```

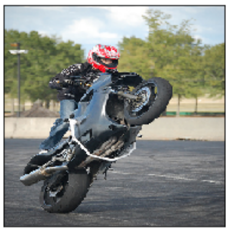


6. Model Evaluation

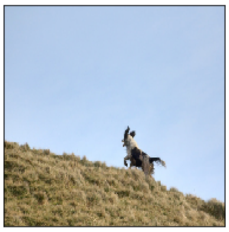
a. Take a random image and generate caption for that image

In [63]:

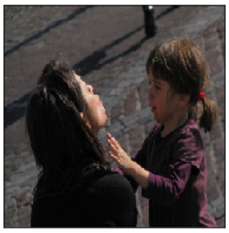
```
index_word = dict([(index,word) for word, index in tokenizer.word_index.items()])
def predict_caption(image):
    '''
    image.shape = (1,4462)
    '''
    in_text = 'startseq'

    for iword in range(maxlen):
        sequence = tokenizer.texts_to_sequences([in_text])[0]
        sequence = pad_sequences([sequence],maxlen)
        yhat = model.predict([image,sequence],verbose=0)
        yhat = np.argmax(yhat)
        newword = index_word[yhat]
        in_text += " " + newword
        if newword == "endseq":
            break
    return(in_text)



npic = 5
npix = 224
target_size = (npix,npix,3)

count = 1
fig = plt.figure(figsize=(10,20))
for jpgfnm, image_feature in zip(fnm_test[:npic],di_test[:npic]):
    ## images
    filename = dir_Flickr_jpg + '/' + jpgfnm
    image_load = load_img(filename, target_size=target_size)
    ax = fig.add_subplot(npic,2,count,xticks=[],yticks=[])
    ax.imshow(image_load)
    count += 1

    ## captions
    caption = predict_caption(image_feature.reshape(1,len(image_feature)))
    ax = fig.add_subplot(npic,2,count)
    plt.axis('off')
    ax.plot()
    ax.set_xlim(0,1)
    ax.set_ylim(0,1)
    ax.text(0,0.5,caption,fontsize=20)
    count += 1

plt.show()
```



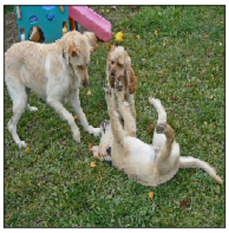
startseq bicycler ride motorcycle on bmx bike endseq



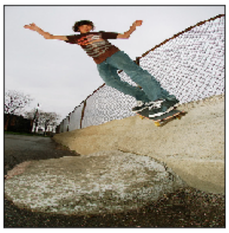
startseq black dog be run through the grass endseq



startseq man in red shirt be stand in front of roped endseq



startseq brown and white dog be run over white shaggy frisbee endseq



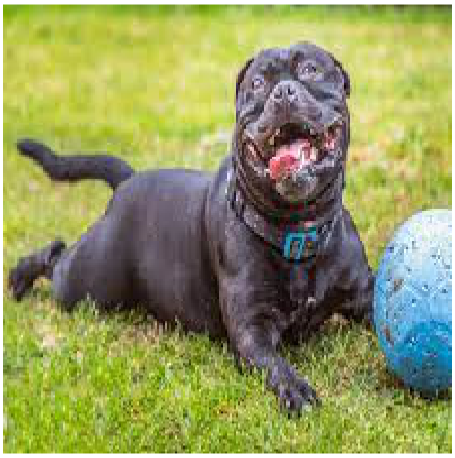
startseq boy in red helmet be move on half pavement endseq

Prediction on random image

In [72]:

```
#Model Evaluation with random google image
filename_t = "../input/new-image/download.jpg"
image_t = load_img(filename_t, target_size=target_size)
image_t
```

Out[72]:



In [75]:

```
# convert the image pixels to a numpy array
image = img_to_array(image_t)
nimage_t = preprocess_input(image)

y_pred_t = modelvgg.predict(nimage_t.reshape( (1,) + nimage_t.shape[:3]))
feature_t = y_pred_t.flatten()

#predicted captions
caption_t = predict_caption(feature_t.reshape(1,len(feature_t)))
caption_t
```

Out[75]:

```
'startseq brown and white dog be run over an shaggy carpet endseq'
```

End notes:

- Due to the limit of resources, we were not able to run it for more number of epochs as colab is timing out or is runnning out of resources.
- We trained for 50 epochs which is giving decent results and we are hopeful that if we increase the number of epochs in training model might have performed better.

In [ ]:

```

```