Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run | Code | Draft Session (2h:18m)

[1]:

```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files un

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved a
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of th
```

```
/kaggle/input/chest-xrays-indiana-university/indiana_projections.csv
/kaggle/input/chest-xrays-indiana-university/indiana_reports.csv
/kaggle/input/chest-xrays-indiana-university/images/images_normalized/349_IM-1697-2001.dcm.png
/kaggle/input/chest-xrays-indiana-university/images/images_normalized/607_IM-2196-1001.dcm.png
/kaggle/input/chest-xrays-indiana-university/images/images_normalized/2832_IM-1249-2001.dcm.png
/kaggle/input/chest-xrays-indiana-university/images/images_normalized/699_IM-2263-2001.dcm.png
```

/kaggle/input/chest-xrays-indiana-university/images/images_normalized/331_IM-2140-2001.dcm.png

[2]:
```
# !kaggle datasets download raddar/chest-xrays-indiana-university
```

[3]:
```
!pip install swifter
!pip install opencv-python
!pip install nltk
```

```
Collecting swifter
  Downloading swifter-1.4.0.tar.gz (1.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.2/1.2 MB 7.9 MB/s eta 0:00:0000:0100:01
  Preparing metadata (setup.py) ... done
Requirement already satisfied: pandas>=1.0.0 in /opt/conda/lib/python3.10/site-packages (from swifter) (2.0.3)
Requirement already satisfied: psutil>=5.6.6 in /opt/conda/lib/python3.10/site-packages (from swifter) (5.9.3)
Requirement already satisfied: dask[dataframe]>=2.10.0 in /opt/conda/lib/python3.10/site-packages (from swifte
r) (2023.12.0)
Requirement already satisfied: tqdm>=4.33.0 in /opt/conda/lib/python3.10/site-packages (from swifter) (4.66.1)
Requirement already satisfied: click>=8.1 in /opt/conda/lib/python3.10/site-packages (from dask[dataframe]>=2.1
0.0->swifter) (8.1.7)
Requirement already satisfied: cloudpickle>=1.5.0 in /opt/conda/lib/python3.10/site-packages (from dask[datafra
me]>=2.10.0->swifter) (2.2.1)
Requirement already satisfied: fsspec>=2021.09.0 in /opt/conda/lib/python3.10/site-packages (from dask[datafram
e]>=2.10.0->swifter) (2023.12.1)
Requirement already satisfied: packaging>=20.0 in /opt/conda/lib/python3.10/site-packages (from dask[dataframe]
```

Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run | Code | Draft Session (2h:19m)

[4]:

```
import pandas as pd
import cv2
import numpy as np
import os
from glob import glob
import math
import matplotlib.pyplot as plt

# import swifter
import re
import html
import string
import unicodedata
from nltk.tokenize import word_tokenize
```

```
/opt/conda/lib/python3.10/site-packages/scipy/__init__.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.
0 is required for this version of SciPy (detected version 1.24.3
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

[5]:

```
df  =pd.read_csv("/kaggle/input/chest-xrays-indiana-university/indiana_reports.csv")
# df[0:6]
```

Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run | Code | Draft Session (2h:19m)

```
from nltk.tokenize import word_tokenize
```

```
/opt/conda/lib/python3.10/site-packages/scipy/__init__.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.
0 is required for this version of SciPy (detected version 1.24.3
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

+ Code + Markdown

[5]:

```
df  =pd.read_csv("/kaggle/input/chest-xrays-indiana-university/indiana_reports.csv")
# df[0:6]
df.count()
```

[5]:

```
uid           3851
MeSH          3851
Problems      3851
image         3851
indication    3765
comparison    2685
findings      3337
impression    3820
dtype: int64
```

[6]:

```
df['findings'].iloc[0:10].tolist()
```

[6]: ['The cardiac silhouette and mediastinum size are within normal limits. There is no pulmonary edema. There is n

Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run | Code | Draft Session (2h:19m)

[6]:
```
df['findings'].iloc[0:10].tolist()
```

[6]:
```
['The cardiac silhouette and mediastinum size are within normal limits. There is no pulmonary edema. There is no focal consolidation. There are no XXXX of a pleural effusion. There is no evidence of pneumothorax.',
 'Borderline cardiomegaly. Midline sternotomy XXXX. Enlarged pulmonary arteries. Clear lungs. Inferior XXXX XXXX XXXX.',
 nan,
 'There are diffuse bilateral interstitial and alveolar opacities consistent with chronic obstructive lung disease and bullous emphysema. There are irregular opacities in the left lung apex, that could represent a cavitary lesion in the left lung apex.There are streaky opacities in the right upper lobe, XXXX scarring. The cardiomediastinal silhouette is normal in size and contour. There is no pneumothorax or large pleural effusion.',
 'The cardiomediastinal silhouette and pulmonary vasculature are within normal limits. There is no pneumothorax or pleural effusion. There are no focal areas of consolidation. Cholecystectomy clips are present. Small T-spine osteophytes. There is biapical pleural thickening, unchanged from prior. Mildly hyperexpanded lungs.',
 'Heart size and mediastinal contour are within normal limits. There is no focal airspace consolidation or suspicious pulmonary opacity. No pneumothorax or large pleural effusion. Mild degenerative change of the thoracic spine.',
 'The cardiac contours are normal. XXXX basilar atelectasis. The lungs are clear. Thoracic spondylosis. Lower cervical XXXX arthritis.',
 'The heart, pulmonary XXXX and mediastinum are within normal limits. There is no pleural effusion or pneumothorax. There is no focal air space opacity to suggest a pneumonia. There is an interim XXXX cervical spinal fusion partly evaluated.',
 'The XXXX examination consists of frontal and lateral radiographs of the chest. The cardiac silhouette is not enlarged. There has been apparent interval increase in low density convexity at the left cardiophrenic XXXX. Calcified granuloma is again seen in the right upper lobe. There is no consolidation, pleural effusion or pneumothorax.',
 'The cardiomediastinal silhouette is within normal limits for size and contour. The lungs are normally inflated without evidence of focal airspace disease, pleural effusion, or pneumothorax. Stable calcified granuloma within the right upper lung. No acute bone abnormality..']
```

Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run | Code | Draft Session (2h:19m)

[7]:
```
df.shape
```

[7]:
```
(3851, 8)
```

+ Code + Markdown

[8]:
```
df['impression'].unique().shape
```

[8]:
```
(1771,)
```

[9]:
```
df['MeSH'].unique().tolist()[:10]
```

[9]:
```
['normal',
 'Cardiomegaly/borderline;Pulmonary Artery/enlarged',
 'Pulmonary Disease, Chronic Obstructive;Bullous Emphysema;Pulmonary Fibrosis/interstitial;Cicatrix/lung/upper
lobe/left;Opacity/lung/apex/left/irregular;Opacity/lung/upper lobe/right/streaky;Opacity/pulmonary alveoli;Opac
ity/lung/bilateral/interstitial/diffuse',
 'Osteophyte/thoracic vertebrae/multiple/small;Thickening/pleura/apex/bilateral;Lung/hyperdistention/mild',
 'Pulmonary Atelectasis/base;Spondylosis/thoracic vertebrae;Arthritis/cervical vertebrae',
```

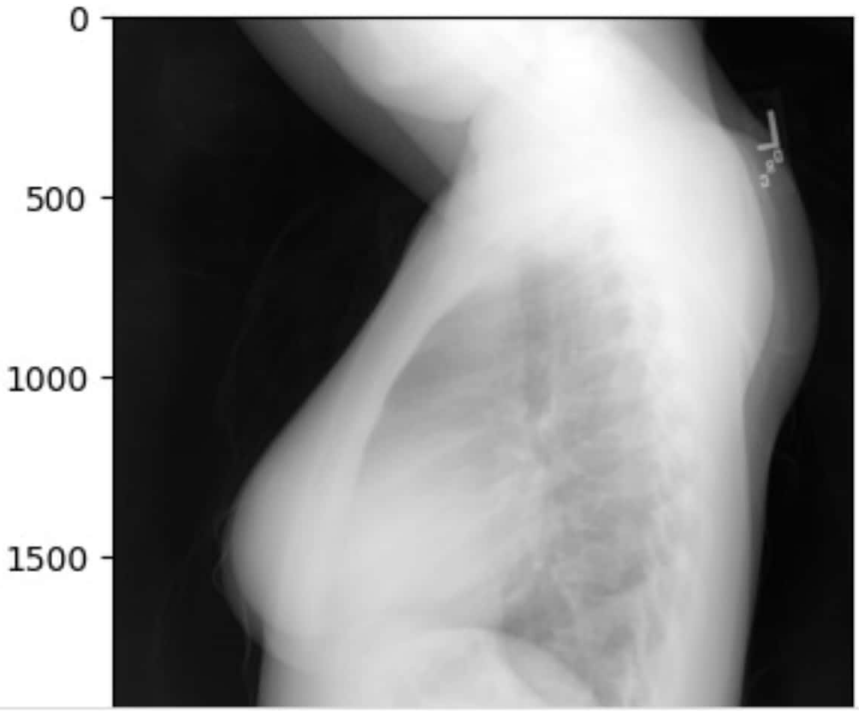
Hamaz_report_generation Draft saved

File Edit View Run Add-ons Help

Cancel Run Code Draft Session (2h:19m)

[10]:
```
img = cv2.imread('/kaggle/input/chest-xrays-indiana-university/images/images_normalized/1_IM-
```

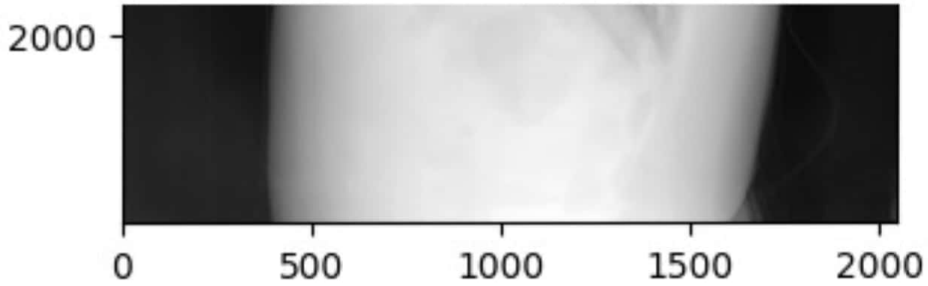
[11]:
```
plt.imshow(img)
plt.show()
```



Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run | Code | Draft Session (2h:19m)



[12]:

```
df2 = pd.read_csv("/kaggle/input/chest-xrays-indiana-university/indiana_projections.csv")
df2.head()
```

[12]:

	uid	filename	projection
0	1	1_IM-0001-4001.dcm.png	Frontal
1	1	1_IM-0001-3001.dcm.png	Lateral
2	2	2_IM-0652-1001.dcm.png	Frontal
3	2	2_IM-0652-2001.dcm.png	Lateral
4	3	3_IM-1384-1001.dcm.png	Frontal

+ Code + Markdown

Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run | Code | Draft Session (2h:20m) | HDD | CPU | RAM | GPU | GPU

4 3 3_IM-1384-1001.dcm.png Frontal

+ Code + Markdown

[13]:
```
df2.projection.unique()
```

[13]:
```
array(['Frontal', 'Lateral'], dtype=object)
```

[14]:
```
import tensorflow
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
```

[15]:
```
def remove_special_chars(text):
    re1 = re.compile(r'  +')
    x1 = text.lower().replace('#39;', "'").replace('amp;', '&').replace('#146;', "'").replace
        'nbsp;', ' ').replace('#36;', '$').replace('\\n', "\n").replace('quot;', "'").replace
        '<br />', "\n").replace('\\"', '"').replace('<unk>', 'u_n').replace(' @.@ ', '.').rep
        ' @-@ ', '-').replace('\\', ' \\ ')
```

Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run | Code | Draft Session (2h:20m)

[15]:

```
def remove_special_chars(text):
    re1 = re.compile(r'  +')
    x1 = text.lower().replace('#39;', "'").replace('amp;', '&').replace('#146;', "'").replace
        'nbsp;', ' ').replace('#36;', '$').replace('\\n', "\n").replace('quot;', "'").replace
        '<br />', "\n").replace('\\"', '"').replace('<unk>', 'u_n').replace(' @.@ ', '.').rep
        ' @-@ ', '-').replace('\\', ' \\ ')
    return re1.sub(' ', html.unescape(x1))


def remove_non_ascii(text):
    """Remove non-ASCII characters from list of tokenized words"""
    return unicodedata.normalize('NFKD', text).encode('ascii', 'ignore').decode('utf-8', 'ign


def to_lowercase(text):
    return text.lower()

def remove_punctuation(text):
    """Remove punctuation from list of tokenized words"""
    translator = str.maketrans('', '', string.punctuation)
    return text.translate(translator)
```

```python
def replace_numbers(text):
    """Replace all interger occurrences in list of tokenized words with textual representatio
    return re.sub(r'\d+', '', text)


def remove_whitespaces(text):
    return text.strip()


def remove_stopwords(words, stop_words):
    """
    :param words:
    :type words:
    :param stop_words: from sklearn.feature_extraction.stop_words import ENGLISH_STOP_WORDS
    or
    from spacy.lang.en.stop_words import STOP_WORDS
    :type stop_words:
    :return:
    :rtype:
    """
    return [word for word in words if word not in stop_words]
```

```
    :rtype:
    """
    return [word for word in words if word not in stop_words]


def stem_words(words):
    """Stem words in text"""
    stemmer = PorterStemmer()
    return [stemmer.stem(word) for word in words]

def lemmatize_words(words):
    """Lemmatize words in text"""

    lemmatizer = WordNetLemmatizer()
    return [lemmatizer.lemmatize(word) for word in words]

def lemmatize_verbs(words):
    """Lemmatize verbs in text"""

    lemmatizer = WordNetLemmatizer()
    return ' '.join([lemmatizer.lemmatize(word, pos='v') for word in words])

def text2words(text):
    return word_tokenize(text)
```

Hamaz_report_generation Draft saved

File Edit View Run Add-ons Help

Cancel Run Code Draft Session (2h:20m)

```
    return ' '.join([lemmatizer.lemmatize(word, pos='v') for word in words])

def text2words(text):
    return word_tokenize(text)

def normalize_text( text):
    text = remove_special_chars(text)
    text = remove_non_ascii(text)
    text = remove_punctuation(text)
    text = to_lowercase(text)
    text = replace_numbers(text)
    #words = text2words(text)
    #stop_words = stopwords.words('english')
    #words = remove_stopwords(words, stop_words)
    #words = stem_words(words)# Either stem ovocar lemmatize
    #words = lemmatize_words(words)
    #words = lemmatize_verbs(words)

    return text

def normalize_corpus(corpus):
    return [normalize_text(t) for t in corpus]
```

Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run | Code | Draft Session (2h:20m) | HDD | CPU | RAM | GPU | GPU

+ Code + Markdown

[16]:
```
df['report'] = df[df.columns[1:]].apply(
    lambda x: ','.join(x.astype(str)),
    axis=1
)
df['report'].head()
```

[16]:
```
0    normal,normal,Xray Chest PA and Lateral,Positi...
1    Cardiomegaly/borderline;Pulmonary Artery/enlar...
2    normal,normal,Xray Chest PA and Lateral,rib pa...
3    Pulmonary Disease, Chronic Obstructive;Bullous...
4    Osteophyte/thoracic vertebrae/multiple/small;T...
Name: report, dtype: object
```

[17]:
```
df['report']
```

[17]:
```
0       normal,normal,Xray Chest PA and Lateral,Positi...
1       Cardiomegaly/borderline;Pulmonary Artery/enlar...
2       normal,normal,Xray Chest PA and Lateral,rib pa...
3       Pulmonary Disease, Chronic Obstructive;Bullous...
4       Osteophyte/thoracic vertebrae/multiple/small;T...
                              ...
3846    Lung/hyperdistention/mild;Diaphragm/bilateral/...
3847    Spine/degenerative Spine Xray Chest PA and Lat...
```

Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run Code Draft Session (2h:20m)

[18]:
```
df['report'] = df['report'].apply(normalize_text)
```

[19]:
```
df['report'] = '<start> '+df['findings']+' <end>'
df.dropna(subset = ['report'], inplace = True)
```

[20]:
```
# import nltk
# nltk.download('punkt')
```

+ Code + Markdown

[21]:
```
num_words = []
for row in df['report'].tolist():
#     print(len(word_tokenize(row)))
    num_words.append(len(word_tokenize(row)))
```

Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run | Code | Draft Session (2h:20m)

[22]:

```
num_words= np.array(num_words)
print("min length              : ", num_words.min())
print("max length              : ", num_words.max())
print("50th percentile length : ", np.percentile(num_words,50))
print("75th percentile length : ", np.percentile(num_words,75))
print("90th percentile length : ", np.percentile(num_words,90))
print("95th percentile length : ", np.percentile(num_words,95))
print("98th percentile length : ", np.percentile(num_words,98))
print("98th percentile length : ", np.percentile(num_words,99))
```

```
min length              :  14
max length              :  193
50th percentile length :  41.0
75th percentile length :  50.0
90th percentile length :  64.0
95th percentile length :  73.0
98th percentile length :  84.27999999999975
98th percentile length :  94.63999999999987
```

Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run | Code | Draft Session (2h:20m)

```
95th percentile length :  73.0
98th percentile length :  84.27999999999975
98th percentile length :  94.63999999999987
```

[23]:

```
vocab_size = 10000
max_len = 260

tok = Tokenizer(num_words=vocab_size,  oov_token='UNK', )
tok.fit_on_texts(df['report'].tolist())
```

[24]:

```
df2 = df2[df2['projection']=='Frontal']
```

[25]:

```
df  =pd.merge(df,df2,  on=['uid'])
```

[26]:

```
df=pd.merge(df,df2,  on=['uid'])
```

[26]:
```
df.head()
```

[26]:

	uid	MeSH	Problems	image	indication	comparison	findings	impressi
0	1	normal	normal	Xray Chest PA and Lateral	Positive TB test	None.	The cardiac silhouette and mediastinum size ar...	Normal ches XX
1	2	Cardiomegaly/borderline;Pulmonary Artery/enlarged	Cardiomegaly;Pulmonary Artery	Chest, 2 views, frontal and lateral	Preop bariatric surgery.	None.	Borderline cardiomegaly. Midline sternotomy XX...	No ac pulmon findin
2	4	Pulmonary Disease, Chronic Obstructive;Bullous...	Pulmonary Disease, Chronic Obstructive;Bullous...	PA and lateral views of the chest XXXX, XXXX	XXXX-year-old XXXX with XXXX.	None available	There are diffuse bilateral interstitial and a...	1. Bull emphysema a intersti fibros

Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run | Code | Draft Session (2h:20m) | HDD | CPU | RAM | GPU | GPU

[27]:

```
class det_gen(tensorflow.keras.utils.Sequence):
    'Generates data from a Dataframe'
    def __init__(self,df, tok, max_len,images_path, dim=(256,256), batch_size=8):
        self.df=df
        self.dim = dim
        self.images_path = images_path
        self.tok= tok
        self.max_len = max_len
        self.batch_size = batch_size
        self.nb_iteration = math.ceil((self.df.shape[0])/self.batch_size)

    def __len__(self):
        'Denotes the number of batches per epoch'
        return self.nb_iteration

    def on_epoch_end(self):
        'Updates indexes after each epoch'
        self.df=self.df.sample(frac=1)

    def load_img(self, img_path):

        img = cv2.imread(img_path)
```

```
        img =cv2.resize(img,(self.dim))


        return img


    def __getitem__(self, index):
        'Generate one batch of data'

        indicies = list(range(index*self.batch_size, min((index*self.batch_size)+self.batch

        images = []
        for img_path in self.df['filename'].iloc[indicies].tolist():
            img = self.load_img(os.path.join(self.images_path,img_path))
            images.append(img)



        x_batch = self.df['report'].iloc[indicies].tolist()

        x_batch_input = [sample[:-len(" <end>")] for sample in x_batch]

        x_batch_gt = [sample[len(" <start>"): ] for sample in x_batch]
```

Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run | Code | Draft Session (2h:21m)

```
        x_batch_input = [sample[:-len(" <end>")] for sample in x_batch]

        x_batch_gt = [sample[len(" <start>"): ] for sample in x_batch]


        x_batch_input = np.array(pad_sequences( self.tok.texts_to_sequences (x_batch_input),
                          maxlen=self.max_len-1 ,
                          padding='post',
                          truncating='post'))

        x_batch_gt = np.array(pad_sequences( self.tok.texts_to_sequences (x_batch_gt),
                          maxlen=self.max_len-1 ,
                          padding='post',
                          truncating='post'))




        return [np.array(images), np.array(x_batch_input)] , np.array(x_batch_gt)
```

Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run | Code | Draft Session (2h:21m)

[28]:
```
validation_split= 0.2
df = df.sample(frac=1)
df_train = df.iloc[:-int(df.shape[0]*validation_split)]
df_val   = df.iloc[-int(df.shape[0]*validation_split):]
```

[29]:
```
images_path = "/kaggle/input/chest-xrays-indiana-university/images/images_normalized/"
train_dataloader =  det_gen(df_train, tok, max_len,images_path)
val_dataloader =  det_gen(df_val, tok, max_len,images_path)
```

[30]:
```
[X_img,X_report] ,Y = next(enumerate(train_dataloader))[1]
```

[31]:
```
plt.imshow(X_img[0])
plt.show()
```

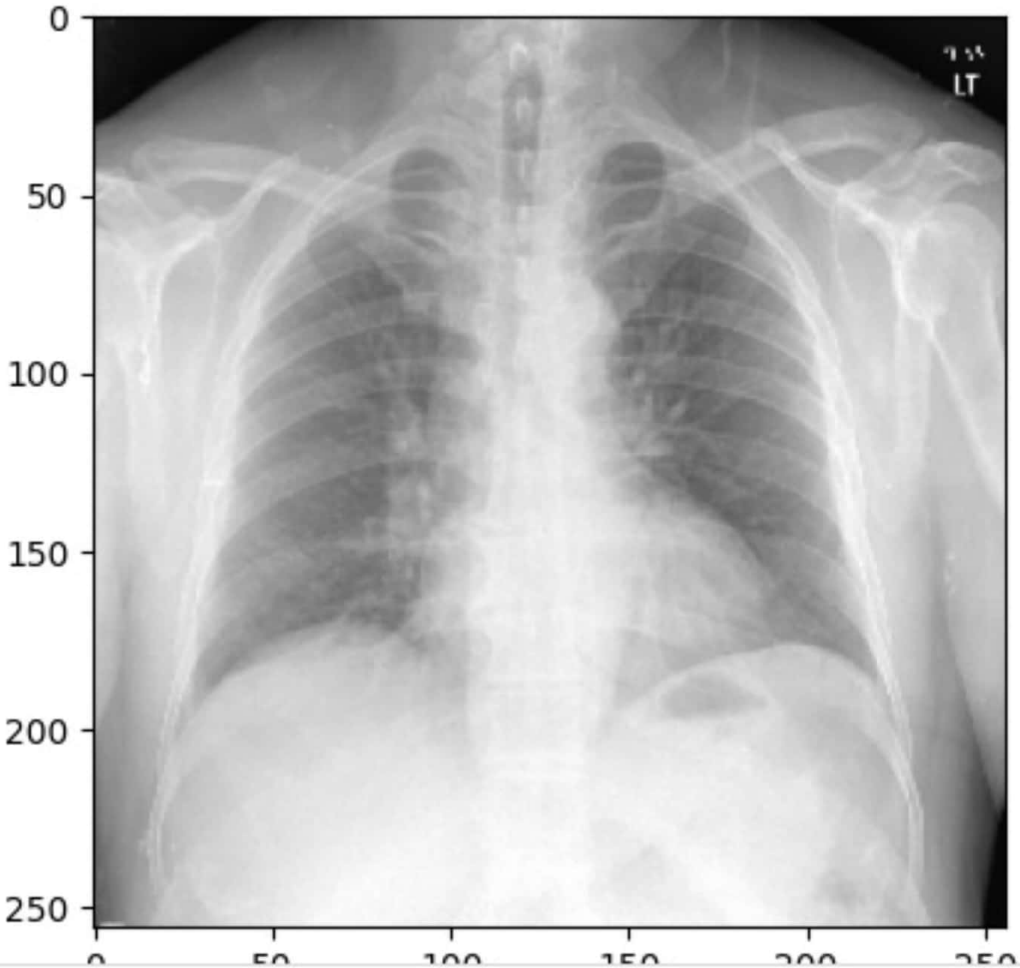
Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run | Code | Draft Session (2h:21m)

[31]:

```
plt.imshow(X_img[0])
plt.show()
```



```
# print("====================")
# print(Y[0])
```

[33]:

```
import tensorflow as tf
from tensorflow.keras import layers
from tensorflow.keras.models import Model
from tensorflow.keras.utils import plot_model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.applications.inception_v3 import InceptionV3
from keras.models import Model
from keras.layers import Flatten, Dense
# from keras.applications import DenseNet121
```

[34]:

```
## Input layers
img_input = layers.Input(shape= (256,256,3))
report_input= layers.Input(shape= (max_len-1,))

## Encoder ######################
```

```
Densenet_model = tf.keras.applications.MobileNetV3Large(
            include_top=False,
            weights="imagenet",
            input_shape=(256,256,3),
        )
number_of_encoder_layers=  len(Densenet_model.layers)

encoder_output = Densenet_model(img_input)
encoder_output = layers.Flatten()(encoder_output)
encoder_output = layers.Dropout(0.2)(encoder_output)
encoder_output = layers.Dense(512,activation='relu')(encoder_output)

##decoder ########################

#layers
gru_layer =  layers.GRU(512, return_sequences=True)
dense_layer= layers.Dense(vocab_size,activation='softmax')
embedding_layer = layers.Embedding(vocab_size, 300, mask_zero=True)
dropout = layers.Dropout(0.2)

# decoder model
embedding_output = embedding_layer(report_input)
gru_output = gru_layer(embedding_output, initial_state=encoder_output )
gru_output = dropout(gru_output)
```

Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run Code Draft Session (2h:21m)

```
# decoder model
embedding_output = embedding_layer(report_input)
gru_output = gru_layer(embedding_output, initial_state=encoder_output )
gru_output = dropout(gru_output)
output  = dense_layer(gru_output)
model = Model([img_input,report_input ],output)

##Inference models ################

#encoder_inference model
encoder_model = Model(img_input,encoder_output)

#decoder_inference model
prev_hidden_state= layers.Input(shape= (512))
report_input2 = layers.Input(shape= (1,))
embedding_output2= embedding_layer(report_input2)
gru_output2 = gru_layer(embedding_output2, initial_state=prev_hidden_state )
gru_output2 = dropout(gru_output2)
output2 = dense_layer(gru_output2)
decoder_model = Model([report_input2,prev_hidden_state],[output2,gru_output2])
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/mobilenet_v3/weights_mobilenet_v3_large_224_1.0_float_no_top_v2.h5
```

Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run | Code | Draft Session (2h:21m)

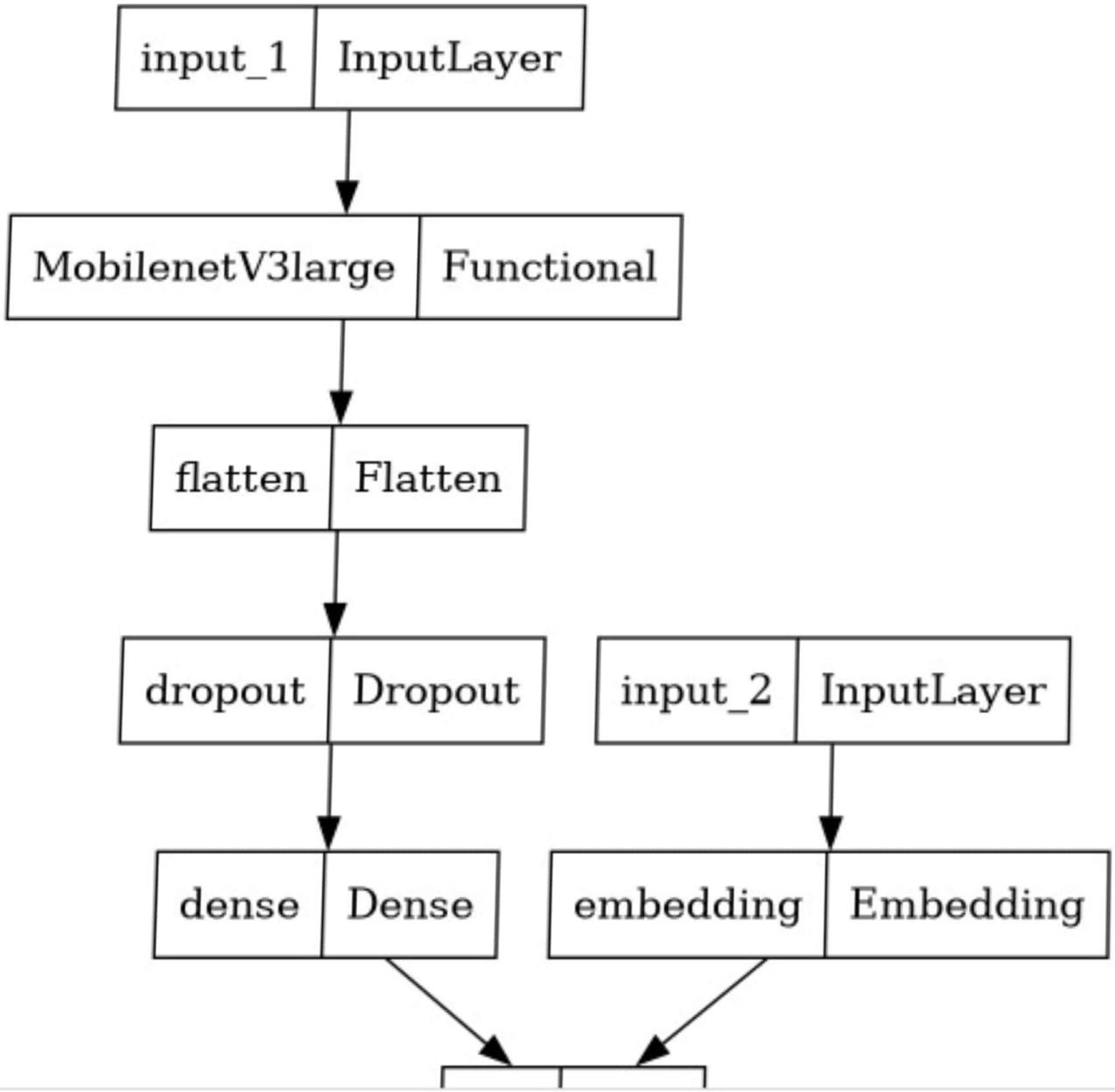
```
Model: "model"
_________________________________________________________________________________________________
 Layer (type)                Output Shape              Param #   Connected to
=================================================================================================
 input_1 (InputLayer)        [(None, 256, 256, 3)]     0         []

 MobilenetV3large (Function  (None, 8, 8, 960)         2996352   ['input_1[0][0]']
 al)

 flatten (Flatten)           (None, 61440)             0         ['MobilenetV3large[0][0]']

 input_2 (InputLayer)        [(None, 259)]             0         []

 dropout (Dropout)           (None, 61440)             0         ['flatten[0][0]']

 embedding (Embedding)       multiple                  3000000   ['input_2[0][0]']

 dense (Dense)               (None, 512)               3145779   ['dropout[0][0]']
                                                       2

 gru (GRU)                   multiple                  1250304   ['embedding[0][0]',
                                                                  'dense[0][0]']

 dropout_1 (Dropout)         multiple                  0         ['gru[0][0]']

 dense_1 (Dense)             multiple                  5130000   ['dropout_1[0][0]']

=================================================================================================
Total params: 43834448 (167.22 MB)
Trainable params: 43810048 (167.12 MB)
Non-trainable params: 24400 (95.31 KB)
_________________________________________________________________________________________________
```

Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run Code Draft Session (2h:21m)

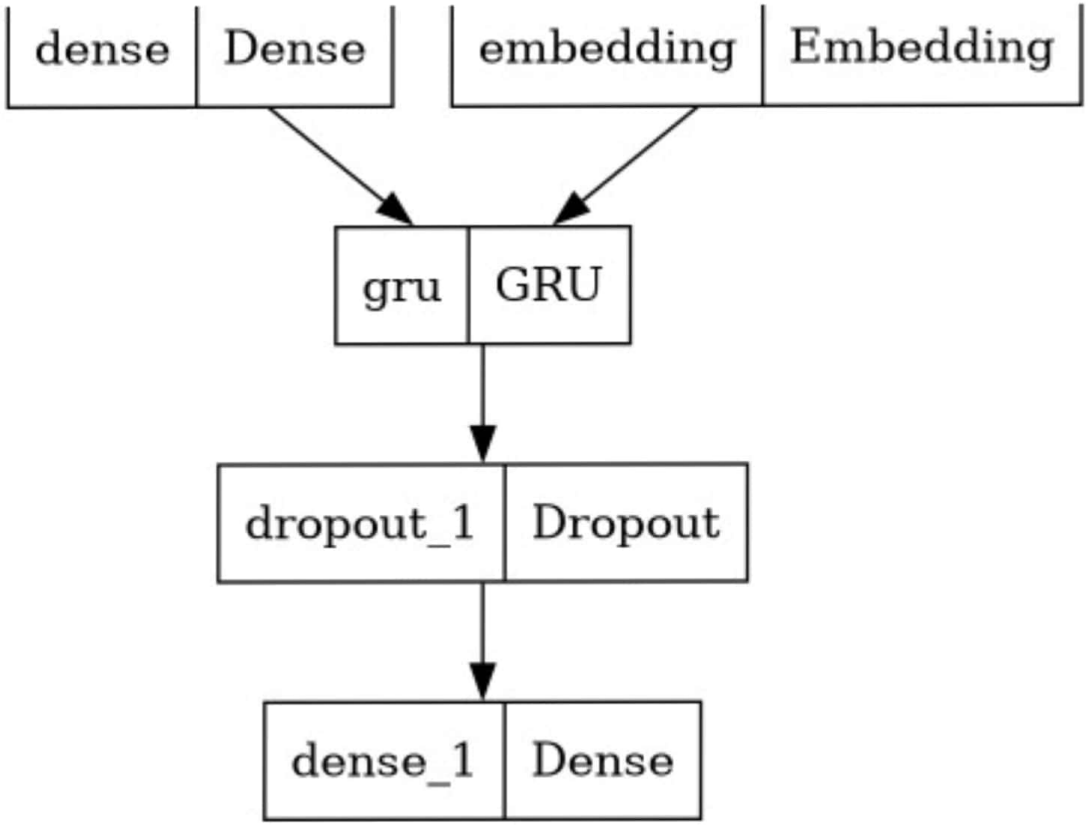
[36]:



Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run Code Draft Session (2h:21m)



+ Code + Markdown

[37]:

```
epochs=38
lr=1e-3
model.compile(loss='sparse_categorical_crossentropy',optimizer=Adam(lr)),
metrics=["accuracy"]
```

Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run | Code | Draft Session (2h:21m)

[37]:
```
epochs=38
lr=1e-3
model.compile(loss='sparse_categorical_crossentropy',optimizer=Adam(lr)),
metrics=["accuracy"]
```

[38]:
```
hist = model.fit_generator( train_dataloader,
                    validation_data = val_dataloader,
                    epochs = epochs
                )
```

```
/tmp/ipykernel_42/2213431721.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a fu
ture version. Please use `Model.fit`, which supports generators.
  hist = model.fit_generator( train_dataloader,
Epoch 1/38
331/331 [==============================] - 281s 693ms/step - loss: 3.5166 - val_loss: 4.0755
Epoch 2/38
331/331 [==============================] - 163s 492ms/step - loss: 2.1840 - val_loss: 2.3510
Epoch 3/38
331/331 [==============================] - 163s 494ms/step - loss: 1.8169 - val_loss: 2.0127
Epoch 4/38
331/331 [==============================] - 164s 494ms/step - loss: 1.5756 - val_loss: 1.8821
Epoch 5/38
```

Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run | Code | Draft Session (2h:21m)

```
331/331 [==============================] - 159s 480ms/step - loss: 0.3705 - val_loss: 4.2488
Epoch 26/38
331/331 [==============================] - 160s 482ms/step - loss: 0.3369 - val_loss: 2.3975
Epoch 27/38
331/331 [==============================] - 160s 482ms/step - loss: 0.3084 - val_loss: 2.3869
Epoch 28/38
331/331 [==============================] - 161s 486ms/step - loss: 0.3115 - val_loss: 2.6628
Epoch 29/38
331/331 [==============================] - 159s 481ms/step - loss: 0.2992 - val_loss: 2.4188
Epoch 30/38
331/331 [==============================] - 161s 485ms/step - loss: 0.3899 - val_loss: 66.3579
Epoch 31/38
331/331 [==============================] - 162s 489ms/step - loss: 0.3771 - val_loss: 2.4591
Epoch 32/38
331/331 [==============================] - 161s 485ms/step - loss: 0.3149 - val_loss: 2.5816
Epoch 33/38
331/331 [==============================] - 161s 485ms/step - loss: 0.2861 - val_loss: 2.4398
Epoch 34/38
331/331 [==============================] - 161s 486ms/step - loss: 0.2632 - val_loss: 2.5134
Epoch 35/38
331/331 [==============================] - 160s 482ms/step - loss: 0.2621 - val_loss: 2.5158
Epoch 36/38
331/331 [==============================] - 160s 482ms/step - loss: 0.2605 - val_loss: 2.5300
Epoch 37/38
331/331 [==============================] - 161s 486ms/step - loss: 0.2632 - val_loss: 2.5735
Epoch 38/38
331/331 [==============================] - 160s 482ms/step - loss: 0.2679 - val_loss: 2.9538
```

+ Code + Markdown

[39]:

```
import numpy as np
from nltk.translate.bleu_score import corpus_bleu
```

Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run Code Draft Session (2h:22m)

[39]:

```
import numpy as np
from nltk.translate.bleu_score import corpus_bleu
```

[40]:

```
def tokens_to_text(tokens,tok,end_token='end'):
    sentence=""
    for token in tokens:
        if token ==0:
            break

        word = tok.index_word[token]

        if word==end_token:
            break

        sentence+= word+" "

    sentence = sentence.strip()
```

Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run | Code | Draft Session (2h:22m)

```
        word = np.argmax(word_probs)

        try:
            if tok.index_word[word]==end_token:
                break
        except:
            pass

        words.append(word)

    words = tokens_to_text(words,tok)
    return words
```

[41]:

```
import cv2
image = cv2.imread("/kaggle/input/chest-xrays-indiana-university/images/images_normalized/32_
# print(image)
import matplotlib.pyplot as plt
%matplotlib inline
plt.imshow(image)
image = cv2.resize(image,(256,256))
image.shape
```

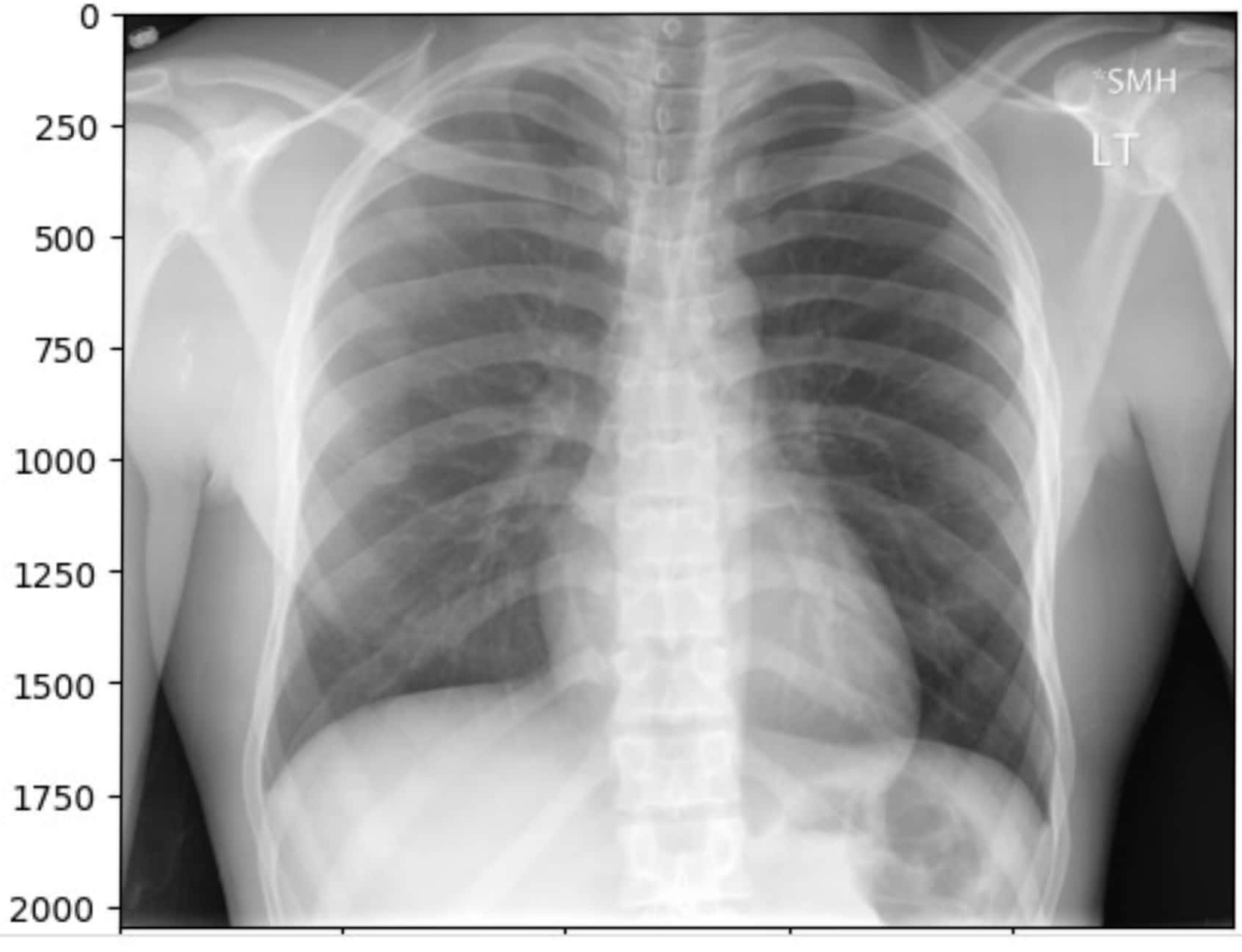
Hamaz_report_generation

File Edit View Run Add-ons Help

Run All | Code | Draft Session (2h:49m) | HDD | CPU | RAM | GPU | GPU

```
(256,256))
```

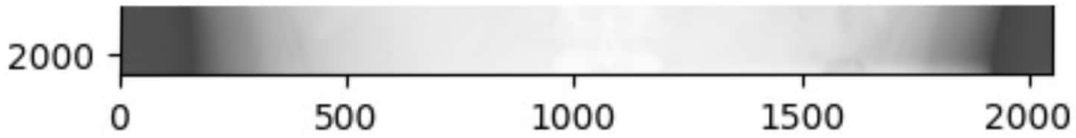
[52]: (256, 256, 3)



Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run | Code | Draft Session (2h:22m)



[42]:

```
pred_sentence = greedy_inference(image, tok,encoder_model, decoder_model,max_len,
                                 start_token="start",end_token='end',decoder
```

```
1/1 [==============================] - 1s 1s/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 19ms/step
```

Hamaz_report_generation

File Edit View Run Add-ons Help

Run All | Code | Draft Session (2h:49m)

```
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 22ms/step
```

```
print(pred_sentence)
```

```
the cardiomediastinal silhouette is within normal limits for appearance no focal areas of pulmonary consolidati
on no pneumothorax no pleural effusion the thoracic spine appears intact no acute displaced rib fractures
```

+ Code + Markdown

Hamaz_report_generation

File Edit View Run Add-ons Help

Cancel Run | Code | Draft Session (2h:22m) | HDD | CPU | RAM | GPU | GPU

```
pred_sentences = []
Gt_sentences = []
for index, (X,Y) in enumerate(data_loader_iterator):
    for img,_,sample_y in zip(X[0],X[1],Y):

        if inference_type=='greedy':
            pred_sentence = greedy_inference(img, tok,encoder_model, decoder_model,max_le
                                             start_token="start",end_token='end',decoder_

        GT_sentence   = tokens_to_text(sample_y,tok)

        pred_sentences.append(pred_sentence)
        Gt_sentences.append(GT_sentence)

    if index == data_loader.nb_iteration -1:
        break
    print("Done with batch number: {} ", index)

return  pred_sentences,Gt_sentences
```

Hamaz_report_generation

Draft saved

File Edit View Run Add-ons Help

Run All | Code | Draft Session (2h:50m) | HDD | CPU | RAM | GPU | GPU

[*]:
```
!pip install rouge==1.0.1
```

[48]:
```
from rouge import Rouge
rouge = Rouge()
rouge.get_scores(pred_sentences, GT_sentences, avg = True)
```

[48]:
```
{'rouge-1': {'r': 0.40668901835710886,
  'p': 0.4053294761920105,
  'f': 0.38384439496024625},
 'rouge-2': {'r': 0.1463346125007915,
  'p': 0.13946746528346476,
  'f': 0.1342126259015835},
 'rouge-l': {'r': 0.29996576676888836,
  'p': 0.2983126618192737,
  'f': 0.2824709369566964}}
```

Hamaz_report_generation Draft saved

File Edit View Run Add-ons Help

Run All Code Draft Session (2h:51m)

[50]:
```
def calculate_bleu_evaluation(GT_sentences, predicted_sentences):
    BLEU_1 = corpus_bleu(GT_sentences, predicted_sentences, weights=(1,0, 0 ,0 ))
    BLEU_2 = corpus_bleu(GT_sentences, predicted_sentences, weights=(0.5, 0.5, 0, 0))
    BLEU_3 = corpus_bleu(GT_sentences, predicted_sentences, weights=(0.33, 0.33, 0.33, 0))
    BLEU_4 = corpus_bleu(GT_sentences, predicted_sentences, weights=(0.25, 0.25, 0.25, 0.25)
    return BLEU_1,BLEU_2,BLEU_3,BLEU_4
```

[51]:
```
BLEU_1,BLEU_2,BLEU_3,BLEU_4 =  evaluate_from_dataloader(val_dataloader,tok,encoder_model, de
print(BLEU_4)
print('-------')
print(BLEU_3),
print('-------')
print(BLEU_2)
print('-------'),
print(BLEU_1)
```

```
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
```

Hamaz_report_generation Draft saved

File Edit View Run Add-ons Help

Run All Code Draft Session (2h:46m)

```
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 26ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - ETA: 0s0.5605890772556354
-------
0.46581225873008997
-------
0.31426011353832484
-------
0.09875941896112082
```

+ Code + Markdown