error_analysis

July 9, 2020

[24]:
```
import warnings
warnings.filterwarnings('ignore')
import pandas as pd
import numpy as np
import random
import string
from tqdm import tqdm
import matplotlib.pyplot as plt
%matplotlib inline
import tensorflow as tf
import matplotlib.image as mpimg
from nltk.translate.bleu_score import sentence_bleu
import pickle
```

[2]:
```
tf.version.VERSION
```

[2]:
```
'2.2.0'
```

[3]:
```
val_images = np.load("val_img.npy",allow_pickle=True)
val_actual = np.load("val_actual.npy",allow_pickle=True)
```

[4]:
```
val_images[0], val_actual[0]
```

[4]:
```
(array(['CXR2978_IM-1367-4001.png', 'CXR2978_IM-1367-1001.png'],
       dtype=object), '<start> no acute findings <end>')
```

[5]:
```
#Unzip the uploaded files, folder names are same for zip file name
tokenizer = pickle.load(open("tokenizer.pkl", 'rb'))
decoder = tf.saved_model.load('decoder/')
encoder = tf.saved_model.load('encoder/')
```

[6]:
```
from tensorflow.keras.applications.inception_v3 import InceptionV3,␣
 ↪preprocess_input
import os
image_features_model = InceptionV3(include_top=False, pooling='avg',␣
 ↪input_shape=(299,299,3))
image_features_model.load_weights("trained_weights-07-0.9102.hdf5")
```

[7]:
```
#Provide image path
image_path = "img/"
```

[8]:
```
max_len_output =80
units = 128
```

[9]:
```
def get_img_tensor(image_path, img_name, model_image):
    img = tf.io.read_file(image_path + str(img_name))
    img = tf.image.decode_jpeg(img, channels=3)
    img = tf.image.resize(img, (299, 299))
    img = preprocess_input(img)
    img_features = model_image(tf.constant(img)[None, :])
    return img_features
```

[10]:
```
def calculate_score(x):
    """Calculates the cumulative score for the length of sentence"""
    return x[1]/len(x[0])

def beam_search(img_name, beam_index = 3):
    """Beam search implementaion takes images as input"""
    hidden =  tf.zeros((1, units))
    img_tensor = tf.convert_to_tensor([get_img_tensor(image_path,img_name[0],␣
 ↪image_features_model),
                                       get_img_tensor(image_path,img_name[1],␣
 ↪image_features_model)])
    img_features = tf.constant(img_tensor)[None, :]
    features_val = encoder(img_features)
    start = [tokenizer.word_index["<start>"]]
    dec_word = [[start, 0.0]]
    while len(dec_word[0][0]) < max_len_output:
        temp = []
        for s in dec_word:
            predictions, hidden = decoder([tf.cast(tf.expand_dims([s[0][-1]],␣
 ↪0), tf.float32), features_val, hidden])

            word_preds = np.argsort(predictions[0])[-beam_index:]
            # Getting the top <beam_index>(n) predictions and creating a
            # new list so as to put them via the model again
            for w in word_preds:
                next_cap, prob = s[0][:], s[1]
                next_cap.append(w)
                prob += predictions[0][w]
                temp.append([next_cap, prob.numpy()])
        dec_word = temp
        # Sorting according to the probabilities scores
        dec_word = sorted(dec_word, reverse=False, key=calculate_score)
        # Getting the top words
```

```
        dec_word = dec_word[-beam_index:]
    dec_word = dec_word[-1][0]
    impression = [tokenizer.index_word[i] for i in dec_word if i !=0]
    result = []

    for i in impression:
        if i != '<end>':
            result.append(i)
        else:
            break

    text = ' '.join(result[1:])
    return result, text
```

[203]:
```
columns = ["idx", "image_1", "image_2", "actual", "predicted", "score"]
df = pd.DataFrame(columns = columns)
for i in tqdm(range(len(val_images))):
    result, text_predicted = beam_search(val_images[i])
    actual = ' '.join([str(elem) for elem in val_actual[i].split()[1:-1]])
    predicted = ' '.join([str(elem) for elem in result[1:]])
    df = df.append(pd.Series([i, val_images[i][0], val_images[i][1], actual,␣
 ↪predicted, sentence_bleu([val_actual[i].split()[1:-1]], result[1:],␣
 ↪weights=(1, 0, 0, 0))], index = columns), ignore_index = True)
df.head(20)
```

```
100%|
   | 399/399 [07:17<00:00,  1.10s/it]
```

[203]:

	idx	image_1	image_2
0	0	CXR2978_IM-1367-4001.png	CXR2978_IM-1367-1001.png
1	1	CXR58_IM-2177-2001.png	CXR58_IM-2177-1001.png
2	2	CXR3953_IM-2021-1002.png	CXR3953_IM-2021-1001.png
3	3	CXR3227_IM-1525-1001.png	CXR3227_IM-1525-2001.png
4	4	CXR2820_IM-1244-1001.png	CXR2820_IM-1244-2001.png
5	5	CXR1029_IM-0022-1001.png	CXR1029_IM-0022-1001.png
6	6	CXR1510_IM-0331-2001.png	CXR1510_IM-0331-1001.png
7	7	CXR979_IM-2466-2001.png	CXR979_IM-2466-1001.png
8	8	CXR3662_IM-1821-1001.png	CXR3662_IM-1821-2001.png
9	9	CXR1303_IM-0199-2001-0001.png	CXR1303_IM-0199-2001-0002.png
10	10	CXR1418_IM-0267-1001.png	CXR1418_IM-0267-2002.png
11	11	CXR1005_IM-0006-1001.png	CXR1005_IM-0006-3003.png
12	12	CXR3477_IM-1690-3001.png	CXR3477_IM-1690-2001.png
13	13	CXR3427_IM-1657-1001.png	CXR3427_IM-1657-2001.png
14	14	CXR1067_IM-0048-2001.png	CXR1067_IM-0048-1001.png
15	15	CXR1953_IM-0621-2001.png	CXR1953_IM-0621-1001.png
16	16	CXR1469_IM-0303-1001.png	CXR1469_IM-0303-2001.png
17	17	CXR886_IM-2400-1002.png	CXR886_IM-2400-1001.png

```
18  18       CXR1701_IM-0462-2001.png       CXR1701_IM-0462-1001.png
19  19       CXR2679_IM-1153-1001.png       CXR2679_IM-1153-2001.png

                                                 actual  \
0                                      no acute findings
1                                       no acute disease
2                  no acute cardiopulmonary abnormality
3                      no acute cardiopulmonary process
4                                       no acute disease
5             no pneumonia heart size normal scoliosis
6                  no acute cardiopulmonary abnormality
7                        negative for acute abnormality
8   chest radiograph no acute radiographic cardiop…
9   right upper lobe mass suspicious for neoplasm …
10  no comparison chest x wellexpanded and clear l…
11                                     no acute findings
12                                      no acute disease
13  there is no evidence of acute cardiopulmonary …
14  no radiographic evidence of acute cardiopulmon…
15  no radiographic evidence of acute cardiopulmon…
16                 no acute cardiopulmonary abnormality
17                                      no acute disease
18                                     no acute findings
19  normal heart size and normal mediastinal conto…

                                               predicted     score
0   no evidence of prior excluded could be identified  0.125000
1                     no cardiopulmonary abnormalities  0.333333
2                       no evidence of pleural disease  0.200000
3                     no cardiopulmonary abnormalities  0.477688
4                           no cardiopulmonary disease  0.666667
5   no evidence for pulmonary nodules of the bone …  0.041667
6                               no acute abnormalities  0.477688
7                          no evidence for consolidation  0.250000
8                              no evidence of primordial  0.118092
9                     no cardiopulmonary abnormalities  0.000000
10                              no evidence for disease  0.007549
11                                     no acute findings  1.000000
12                 no acute cardiopulmonary primordial  0.500000
13                                     no acute findings  0.012210
14                    no cardiopulmonary abnormalities  0.175731
15                    no acute cardiopulmonary process  0.354275
16               no acute abnormalities are identified  0.400000
17                    no cardiopulmonary abnormalities  0.333333
18                                     low lung effusion  0.000000
19                          no cardiopulmonary disease  0.000156
```

[206]:
```
df_poor = df[df['score'] < 0.08]
```

[207]:
```
df_poor.head()
```

[207]:

	idx	image_1	image_2	actual	predicted	score
5	5	CXR1029_IM-0022-1001.png	CXR1029_IM-0022-1001.png	no pneumonia heart size normal scoliosis	no evidence for pulmonary nodules of the bone …	0.041667
9	9	CXR1303_IM-0199-2001-0001.png	CXR1303_IM-0199-2001-0002.png	right upper lobe mass suspicious for neoplasm …	no cardiopulmonary abnormalities	0.000000
10	10	CXR1418_IM-0267-1001.png	CXR1418_IM-0267-2002.png	no comparison chest x wellexpanded and clear l…	no evidence for disease	0.007549
13	13	CXR3427_IM-1657-1001.png	CXR3427_IM-1657-2001.png	there is no evidence of acute cardiopulmonary …	no acute findings	0.012210
18	18	CXR1701_IM-0462-2001.png	CXR1701_IM-0462-1001.png	no acute findings	low lung effusion	0.000000

[208]:
```
df_poor['duplicate'] = np.where(df_poor['image_1']==df_poor['image_2'], 1, 0)
```

[209]:
```
df_poor.head()
```

[209]:

	idx	image_1	image_2	actual	predicted	score	duplicate
5	5	CXR1029_IM-0022-1001.png	CXR1029_IM-0022-1001.png	no pneumonia heart size normal scoliosis	no evidence for pulmonary nodules of the bone …	0.041667	1
9	9	CXR1303_IM-0199-2001-0001.png	CXR1303_IM-0199-2001-0002.png	right upper lobe mass suspicious for neoplasm …	no cardiopulmonary abnormalities	0.000000	0
10	10	CXR1418_IM-0267-1001.png	CXR1418_IM-0267-2002.png	no comparison chest x wellexpanded and clear l…	no evidence for disease	0.007549	0
13	13	CXR3427_IM-1657-1001.png	CXR3427_IM-1657-2001.png	there is no evidence of acute cardiopulmonary …	no acute findings	0.012210	0
18	18	CXR1701_IM-0462-2001.png	CXR1701_IM-0462-1001.png	no acute findings	low lung effusion	0.000000	0

```
13                                  no acute findings  0.012210          0
18                                  low lung effusion  0.000000          0
```

[210]:
```
df_poor['duplicate'].value_counts()
```

[210]:
```
0    139
1     22
Name: duplicate, dtype: int64
```

- There are 22 duplicate image data which predicted score is poor as we already know that these data points we considered as noise and equally split among all the data sets.
- Lets ignore those data points in the prediction and perform a analysis from start

[11]:
```
columns = ["idx", "image_1", "image_2", "actual", "predicted", "score"]
df1 = pd.DataFrame(columns = columns)
for i in tqdm(range(len(val_images))):
    if val_images[i][0] != val_images[i][1]:
        result, text_predicted = beam_search(val_images[i])
        actual = ' '.join([str(elem) for elem in val_actual[i].split()[1:-1]])
        predicted = ' '.join([str(elem) for elem in result[1:]])
        df1 = df1.append(pd.Series([i, val_images[i][0], val_images[i][1],␣
 ↪actual, predicted, sentence_bleu([val_actual[i].split()[1:-1]], result[1:],␣
 ↪weights=(1, 0, 0, 0))], index = columns), ignore_index = True)
df1.head(20)
```

```
100%|
  | 399/399 [09:51<00:00,  1.48s/it]
```

[11]:

	idx	image_1	image_2	\
0	0	CXR2978_IM-1367-4001.png	CXR2978_IM-1367-1001.png	
1	1	CXR58_IM-2177-2001.png	CXR58_IM-2177-1001.png	
2	2	CXR3953_IM-2021-1002.png	CXR3953_IM-2021-1001.png	
3	3	CXR3227_IM-1525-1001.png	CXR3227_IM-1525-2001.png	
4	4	CXR2820_IM-1244-1001.png	CXR2820_IM-1244-2001.png	
5	6	CXR1510_IM-0331-2001.png	CXR1510_IM-0331-1001.png	
6	7	CXR979_IM-2466-2001.png	CXR979_IM-2466-1001.png	
7	8	CXR3662_IM-1821-1001.png	CXR3662_IM-1821-2001.png	
8	9	CXR1303_IM-0199-2001-0001.png	CXR1303_IM-0199-2001-0002.png	
9	10	CXR1418_IM-0267-1001.png	CXR1418_IM-0267-2002.png	
10	11	CXR1005_IM-0006-1001.png	CXR1005_IM-0006-3003.png	
11	12	CXR3477_IM-1690-3001.png	CXR3477_IM-1690-2001.png	
12	13	CXR3427_IM-1657-1001.png	CXR3427_IM-1657-2001.png	
13	14	CXR1067_IM-0048-2001.png	CXR1067_IM-0048-1001.png	
14	15	CXR1953_IM-0621-2001.png	CXR1953_IM-0621-1001.png	
15	16	CXR1469_IM-0303-1001.png	CXR1469_IM-0303-2001.png	
16	17	CXR886_IM-2400-1002.png	CXR886_IM-2400-1001.png	
17	18	CXR1701_IM-0462-2001.png	CXR1701_IM-0462-1001.png	

```
18  19       CXR2679_IM-1153-1001.png       CXR2679_IM-1153-2001.png
19  21       CXR2098_IM-0728-1001.png       CXR2098_IM-0728-2001.png

                                                 actual  \
0                                     no acute findings
1                                      no acute disease
2                  no acute cardiopulmonary abnormality
3                      no acute cardiopulmonary process
4                                      no acute disease
5                  no acute cardiopulmonary abnormality
6                        negative for acute abnormality
7   chest radiograph no acute radiographic cardiop…
8   right upper lobe mass suspicious for neoplasm …
9   no comparison chest x wellexpanded and clear l…
10                                    no acute findings
11                                     no acute disease
12  there is no evidence of acute cardiopulmonary …
13  no radiographic evidence of acute cardiopulmon…
14  no radiographic evidence of acute cardiopulmon…
15                 no acute cardiopulmonary abnormality
16                                     no acute disease
17                                    no acute findings
18  normal heart size and normal mediastinal conto…
19                    no acute cardiopulmonary findings

                                               predicted     score
0   no evidence of prior excluded could be identified  0.125000
1                     no cardiopulmonary abnormalities  0.333333
2                       no evidence of pleural disease  0.200000
3                     no cardiopulmonary abnormalities  0.477688
4                           no cardiopulmonary disease  0.666667
5                               no acute abnormalities  0.477688
6                          no evidence for consolidation  0.250000
7                              no evidence of primordial  0.118092
8                     no cardiopulmonary abnormalities  0.000000
9                               no evidence for disease  0.007549
10                                    no acute findings  1.000000
11                 no acute cardiopulmonary primordial  0.500000
12                                    no acute findings  0.012210
13                    no cardiopulmonary abnormalities  0.175731
14                    no acute cardiopulmonary process  0.354275
15                no acute abnormalities are identified  0.400000
16                    no cardiopulmonary abnormalities  0.333333
17                                    low lung effusion  0.000000
18                          no cardiopulmonary disease  0.000156
19                          no cardiopulmonary disease  0.477688
```

Lets visualize the good results images

[12]:
```
df1_best = df1[df1['score'] > 0.5]
df1_best.head()
```

[12]:
```
    idx                   image_1                   image_2  \
4     4  CXR2820_IM-1244-1001.png  CXR2820_IM-1244-2001.png
10   11  CXR1005_IM-0006-1001.png  CXR1005_IM-0006-3003.png
21   23  CXR3735_IM-1866-2001.png  CXR3735_IM-1866-1001.png
28   30  CXR3204_IM-1513-2001.png  CXR3204_IM-1513-1001.png
44   51   CXR179_IM-0514-2001.png   CXR179_IM-0514-1001.png

                                 actual  \
4                      no acute disease
10                    no acute findings
21                     no acute process
28           no acute pulmonary disease
44  no acute cardiopulmonary abnormality

                               predicted     score
4            no cardiopulmonary disease  0.666667
10                    no acute findings  1.000000
21     no acute cardiopulmonary process  0.750000
28            no acute airspace disease  0.750000
44  no acute cardiopulmonary abnormality  1.000000
```

[13]:
```
df_sorted = df1.sort_values('score', ascending=False)
df_sorted.head()
```

[13]:
```
     idx                   image_1                   image_2  \
44    51   CXR179_IM-0514-2001.png   CXR179_IM-0514-1001.png
10    11  CXR1005_IM-0006-1001.png  CXR1005_IM-0006-3003.png
56    64  CXR2658_IM-1140-2001.png  CXR2658_IM-1140-1001.png
348  392  CXR1780_IM-0509-2001.png  CXR1780_IM-0509-1001.png
21    23  CXR3735_IM-1866-2001.png  CXR3735_IM-1866-1001.png

                                    actual  \
44     no acute cardiopulmonary abnormality
10                     no acute findings
56   no acute cardiopulmonary abnormalities
348    no acute cardiopulmonary abnormality
21                      no acute process

                                predicted  score
44   no acute cardiopulmonary abnormality   1.00
10                    no acute findings   1.00
56       no acute cardiopulmonary disease   0.75
```

```
348      no acute cardiopulmonary process   0.75
21       no acute cardiopulmonary process   0.75
```

[14]:
```
len(df_sorted[df_sorted['score'] < 0.08])
```

[14]:
```
139
```

[15]:
```
139/354
```

[15]:
```
0.3926553672316384
```

39% of the data seems having poor bleu score. Lets take the those data points and do some analysis

[16]:
```
df_poor = df_sorted[df_sorted['score'] < 0.08]
df_poor.head()
```

[16]:
```
     idx                    image_1                    image_2  \
301  341  CXR1562_IM-0367-2001.png  CXR1562_IM-0367-1001.png
26    28   CXR219_IM-0799-2001.png   CXR219_IM-0799-1001.png
54    62  CXR1485_IM-0313-1001.png  CXR1485_IM-0313-2001.png
46    53   CXR594_IM-2187-1001.png   CXR594_IM-2187-2001.png
304  344   CXR300_IM-1385-1001.png   CXR300_IM-1385-1002.png

                                              actual  \
301     negative for acute cardiopulmonary abnormality
26   no x evidence of pulmonary metastatic disease …
54   unchanged platelike bibasilar opacities most r…
46   borderline cardiomegaly ageindeterminate chron…
304  changes of chronic lung disease with no acute …

                                          predicted     score
301  no acute subsegmental streaky airways pulmonar…  0.071429
26                      no evidence for consolidation  0.067668
54   low lung characterized within the body acute f…  0.067032
46   low lung sequela of the heart this is within n…  0.066667
304                          no acute abnormalities   0.064648
```

[17]:
```
df_poor['actual_count'] = df_poor['actual'].astype(str).str.split().
↪apply(lambda x: 0 if x==None else len(x))
df_poor.head()
```

[17]:
```
     idx                    image_1                    image_2  \
301  341  CXR1562_IM-0367-2001.png  CXR1562_IM-0367-1001.png
26    28   CXR219_IM-0799-2001.png   CXR219_IM-0799-1001.png
54    62  CXR1485_IM-0313-1001.png  CXR1485_IM-0313-2001.png
46    53   CXR594_IM-2187-1001.png   CXR594_IM-2187-2001.png
304  344   CXR300_IM-1385-1001.png   CXR300_IM-1385-1002.png
```

```
                                                  actual  \
301     negative for acute cardiopulmonary abnormality
26   no x evidence of pulmonary metastatic disease …
54   unchanged platelike bibasilar opacities most r…
46   borderline cardiomegaly ageindeterminate chron…
304  changes of chronic lung disease with no acute …

                                              predicted     score  actual_count
301  no acute subsegmental streaky airways pulmonar…  0.071429             5
26                       no evidence for consolidation  0.067668            12
54   low lung characterized within the body acute f…  0.067032            14
46   low lung sequela of the heart this is within n…  0.066667             9
304                            no acute abnormalities  0.064648            10
```

[18]:
```
df_poor.shape
```

[18]:
```
(139, 7)
```

[19]:
```
df_poor['actual_count'].describe()
```

[19]:
```
count    139.000000
mean      17.532374
std       14.668957
min        2.000000
25%        5.500000
50%       14.000000
75%       25.500000
max      110.000000
Name: actual_count, dtype: float64
```

- minimum is 2 and the maximum word count is 110. we have used the max_len as 80 in our prediction lets ignore those word counts which are greater than 80.

[20]:
```
df_poor = df_poor[df_poor['actual_count'] < 80]
df_poor.shape
```

[20]:
```
(138, 7)
```

Lets take each data point and do the analysis

[149]:
```
k=random.choice(df_poor.index.tolist())
img_nms = [df_poor["image_1"][k], df_poor["image_2"][k]]
fig, axs = plt.subplots(1, len(img_nms), figsize = (10,10), tight_layout=True)
count = 0
for img, subplot in zip(img_nms, axs.flatten()):
    img_=mpimg.imread(image_path+img)
    imgplot = axs[count].imshow(img_, cmap = 'bone')
```

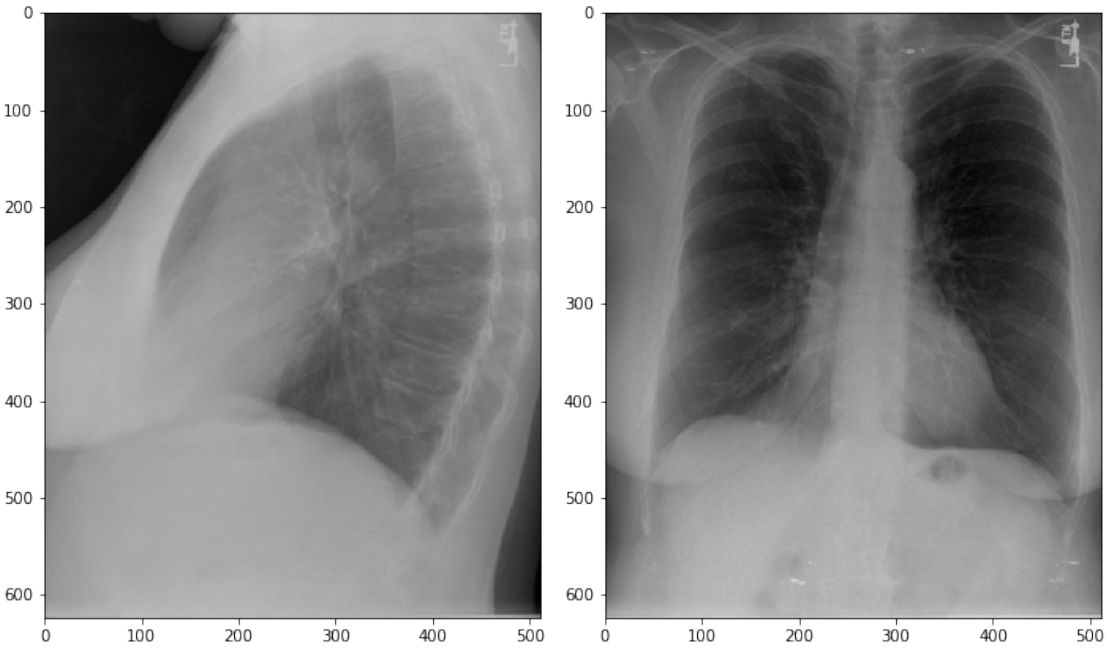
```
    count +=1
plt.show()
print("Score:", df_poor["score"][k])
print("Actual:", df_poor["actual"][k])
print("Predicted:", df_poor["predicted"][k])
print("word count:", df_poor["actual_count"][k])
```


```
Score: 0.0004681758116527773
Actual: stable normal cardiac size and contour unremarkable mediastinal
silhouette normal pulmonary and interstitium lungs clear no airspace disease
pleural effusion or pneumothorax no activeacute cardiopulmonary disease
Predicted: no cardiopulmonary disease
word count: 26
```

- word length is 26 and there is a word overlap **"activeacute"** in actual value could not find any image issue
- Predicted word gives the partial meaning from the actual not a poor prediction.
- we get the poor value because Bleu score does not accounts the meaning

[150]:
```
import random
k=random.choice(df_poor.index.tolist())
img_nms = [df_poor["image_1"][k], df_poor["image_2"][k]]
fig, axs = plt.subplots(1, len(img_nms), figsize = (10,10), tight_layout=True)
count = 0
for img, subplot in zip(img_nms, axs.flatten()):
    img_=mpimg.imread(image_path+img)
```

```
    imgplot = axs[count].imshow(img_, cmap = 'bone')
    count +=1
plt.show()
print("Score:", df_poor["score"][k])
print("Actual:", df_poor["actual"][k])
print("Predicted:", df_poor["predicted"][k])
print("word count:", df_poor["actual_count"][k])
```

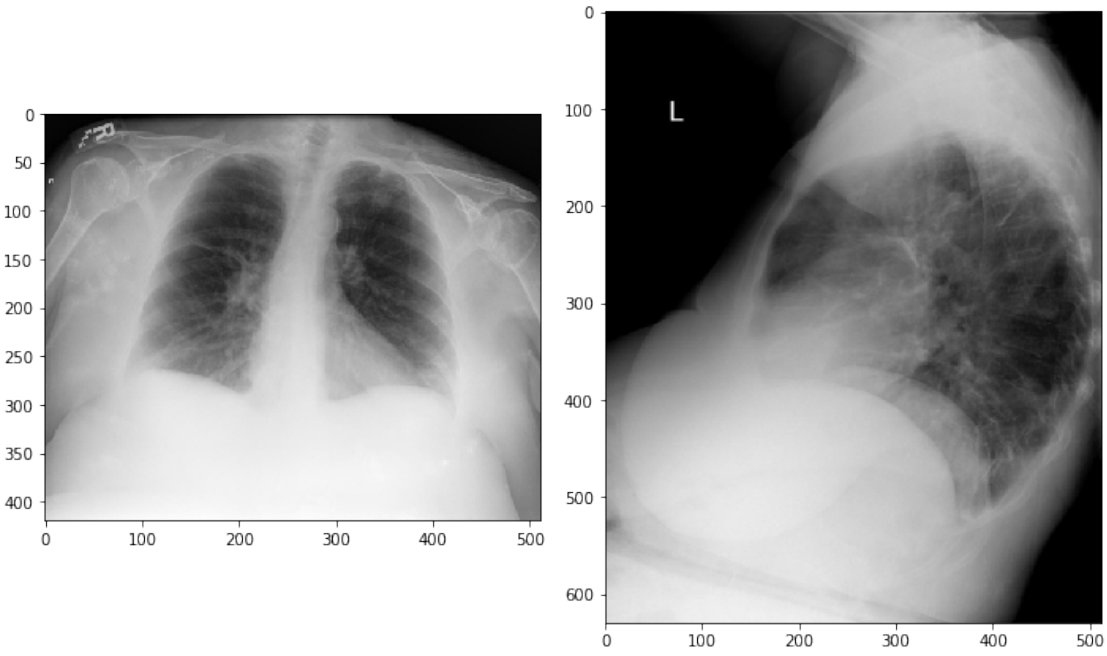



```
Score: 0.0
Actual: small right juxtahilar opacity may represent infiltrate in the setting
of followup chest x is recommended at an appropriate interval following
treatment to document
Predicted: no acute abnormalities
word count: 24
```

- word count is 24, No error in actual word. still cant find any image wise pattern issue
- predicted word is poor did not give any similar meanings

[151]:
```
import random
k=random.choice(df_poor.index.tolist())
img_nms = [df_poor["image_1"][k], df_poor["image_2"][k]]
fig, axs = plt.subplots(1, len(img_nms), figsize = (10,10), tight_layout=True)
count = 0
for img, subplot in zip(img_nms, axs.flatten()):
    img_=mpimg.imread(image_path+img)
    imgplot = axs[count].imshow(img_, cmap = 'bone')
```

```
    count +=1
plt.show()
print("Score:", df_poor["score"][k])
print("Actual:", df_poor["actual"][k])
print("Predicted:", df_poor["predicted"][k])
print("word count:", df_poor["actual_count"][k])
```

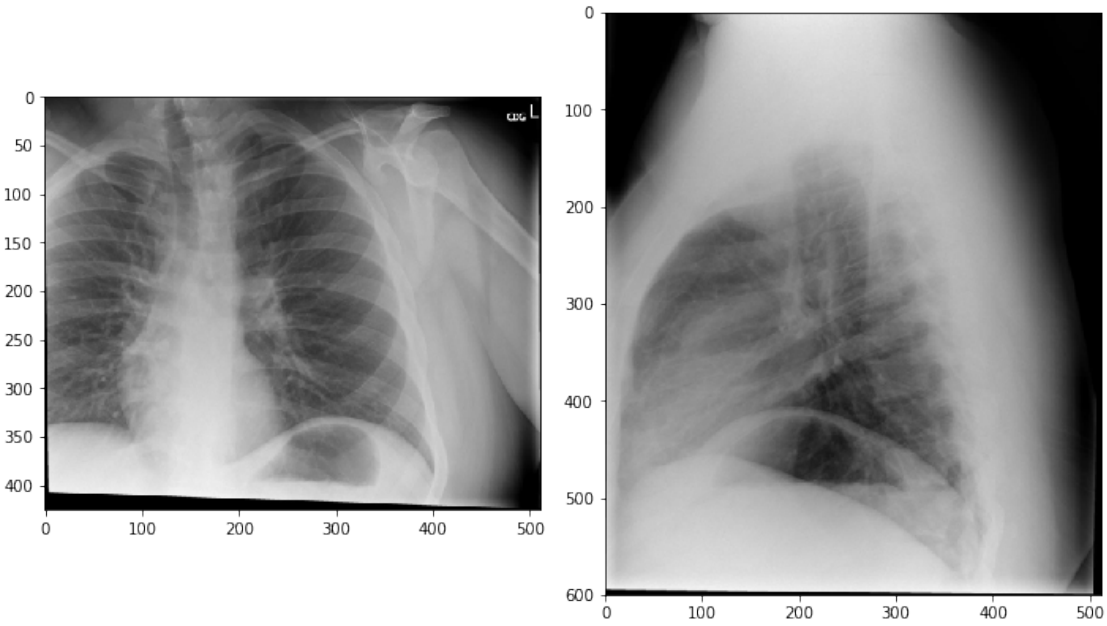



```
Score: 0.0
Actual: no focal air space consolidation nodular opacity at the left apex may be
exaggerated by overlapping bone silhouettes chest may provide further evaluation
if warranted
Predicted: low lung features without acute follow critical result of thoracic
spine portacatheter
word count: 25
```

- word count is 25, no error in actual word, no image patter issue could be identified
- predicted is poor did not give any meaning from actual.

[153]:
```
import random
k=random.choice(df_poor.index.tolist())
img_nms = [df_poor["image_1"][k], df_poor["image_2"][k]]
fig, axs = plt.subplots(1, len(img_nms), figsize = (10,10), tight_layout=True)
count = 0
for img, subplot in zip(img_nms, axs.flatten()):
    img_=mpimg.imread(image_path+img)
    imgplot = axs[count].imshow(img_, cmap = 'bone')
```

```
    count +=1
plt.show()
print("Score:", df_poor["score"][k])
print("Actual:", df_poor["actual"][k])
print("Predicted:", df_poor["predicted"][k])
print("word count:", df_poor["actual_count"][k])
```



```
Score: 0.04632230081520103
Actual: chest no visible active cardiopulmonary disease left hip advanced
posttraumatic osteoarthritis
Predicted: no cardiopulmonary abnormalities
word count: 11
```

- word count is 11, No error in actual word, images is not perfectlly captured when we compare with other.
- Prediction gives same meaning issue with the Bleu score.

As we can see the bleu score which is having value greater than 0 actually gives the partial meaning of actual which considered as good prediction lets take bleu score which are 0.
Another finding is when we have word more than 20 word give 0 values. which shows that our model did not perform well for longer sentence. lets consider word lesser than 20.

[21]:
```
df_poor_zero = df_poor[df_poor['score'] == 0]
df_poor_zero.head()
```

[21]:
```
      idx                    image_1                    image_2  \
49     56  CXR2716_IM-1181-1001.png  CXR2716_IM-1181-2001.png
352   397  CXR1013_IM-0013-1001.png  CXR1013_IM-0013-2001.png
184   211  CXR1304_IM-0199-2001.png  CXR1304_IM-0199-1001.png
204   232   CXR501_IM-2120-2001.png   CXR501_IM-2120-1001.png
203   230  CXR2943_IM-1343-1001.png  CXR2943_IM-1343-4004.png

                                              actual  \
49   right lower lobe airspace disease with bilater…
352  stable mild cardiomegaly without acute cardiop…
184                                     normal chest
204  there is minimal streaky opacity in the poster…
203  mild left costophrenic blunting basilar pleura…

                                     predicted  score  actual_count
49              no cardiopulmonary abnormalities    0.0             9
352  no evidence of pulmonary venous hypertension    0.0             7
184                             no acute findings    0.0             2
204                 low lung features consistent    0.0            24
203             no cardiopulmonary abnormalities    0.0            42
```

[22]:
```
df_poor_zero = df_poor_zero[df_poor_zero['actual_count'] < 20]
df_poor_zero.head()
```

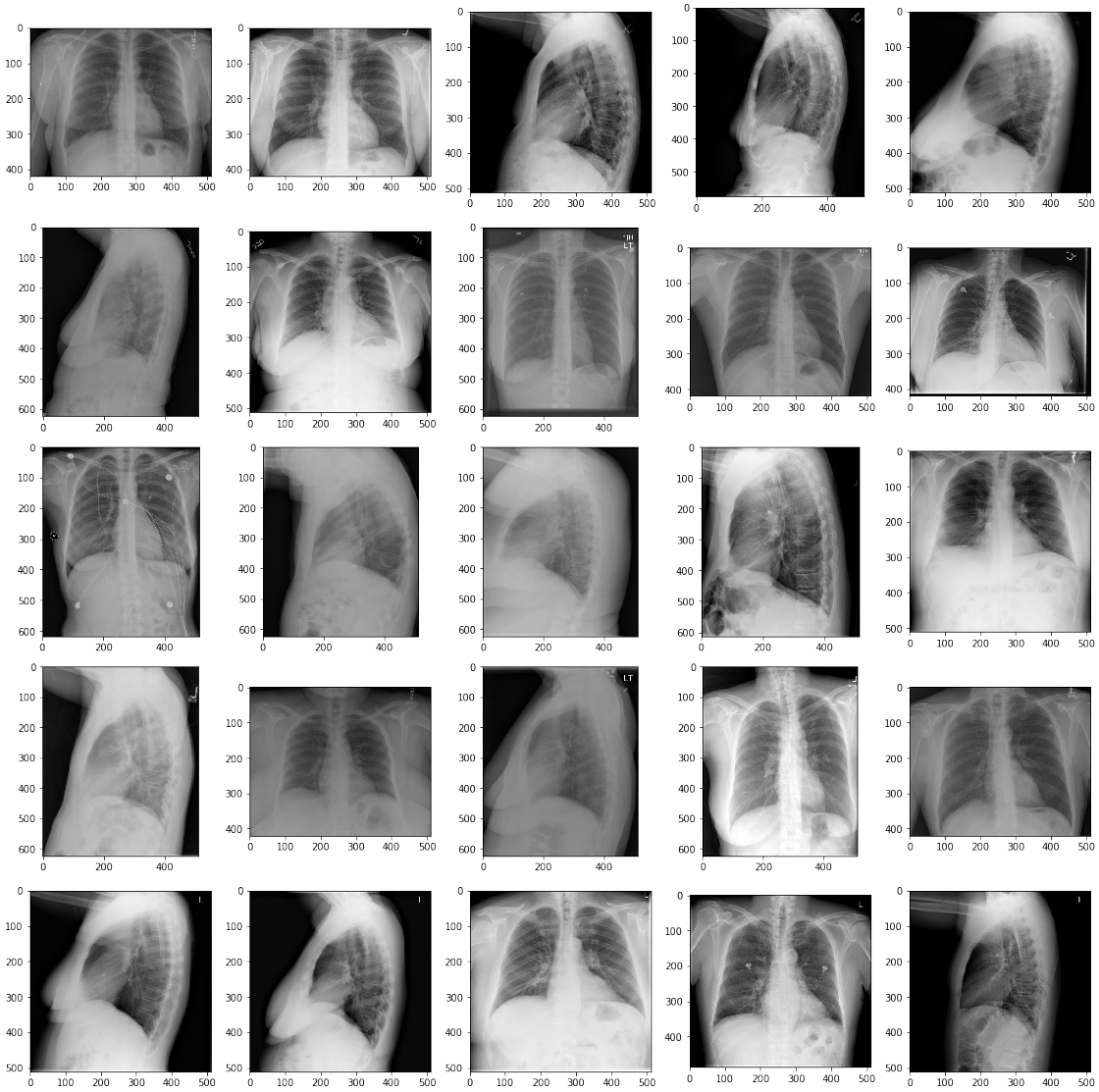
[22]:
```
      idx                        image_1                        image_2  \
49     56       CXR2716_IM-1181-1001.png       CXR2716_IM-1181-2001.png
352   397       CXR1013_IM-0013-1001.png       CXR1013_IM-0013-2001.png
184   211       CXR1304_IM-0199-2001.png       CXR1304_IM-0199-1001.png
208   236      CXR2922_IM-1325-12012.png       CXR2922_IM-1325-1001.png
8       9  CXR1303_IM-0199-2001-0001.png  CXR1303_IM-0199-2001-0002.png

                                              actual  \
49   right lower lobe airspace disease with bilater…
352  stable mild cardiomegaly without acute cardiop…
184                                     normal chest
208  hyperinflated lungs air trapping versus inspir…
8    right upper lobe mass suspicious for neoplasm …

                                     predicted  score  actual_count
49              no cardiopulmonary abnormalities    0.0             9
352  no evidence of pulmonary venous hypertension    0.0             7
184                             no acute findings    0.0             2
208              low lung features are elevation    0.0             6
8               no cardiopulmonary abnormalities    0.0            19
```

[166]:
```
df_poor_zero.shape
```

[166]: (62, 7)

[175]:
```
print("==== Displaying best result random 25 patient X-Ray 1st image ====")
fig, axs = plt.subplots(5, 5, figsize = (16,16), tight_layout=True)
for row, subplot in zip(df1_best[0:25].itertuples(), axs.flatten()):
    img=mpimg.imread(image_path+row.image_1)
    subplot.imshow(img, cmap = 'bone')
plt.show()
```

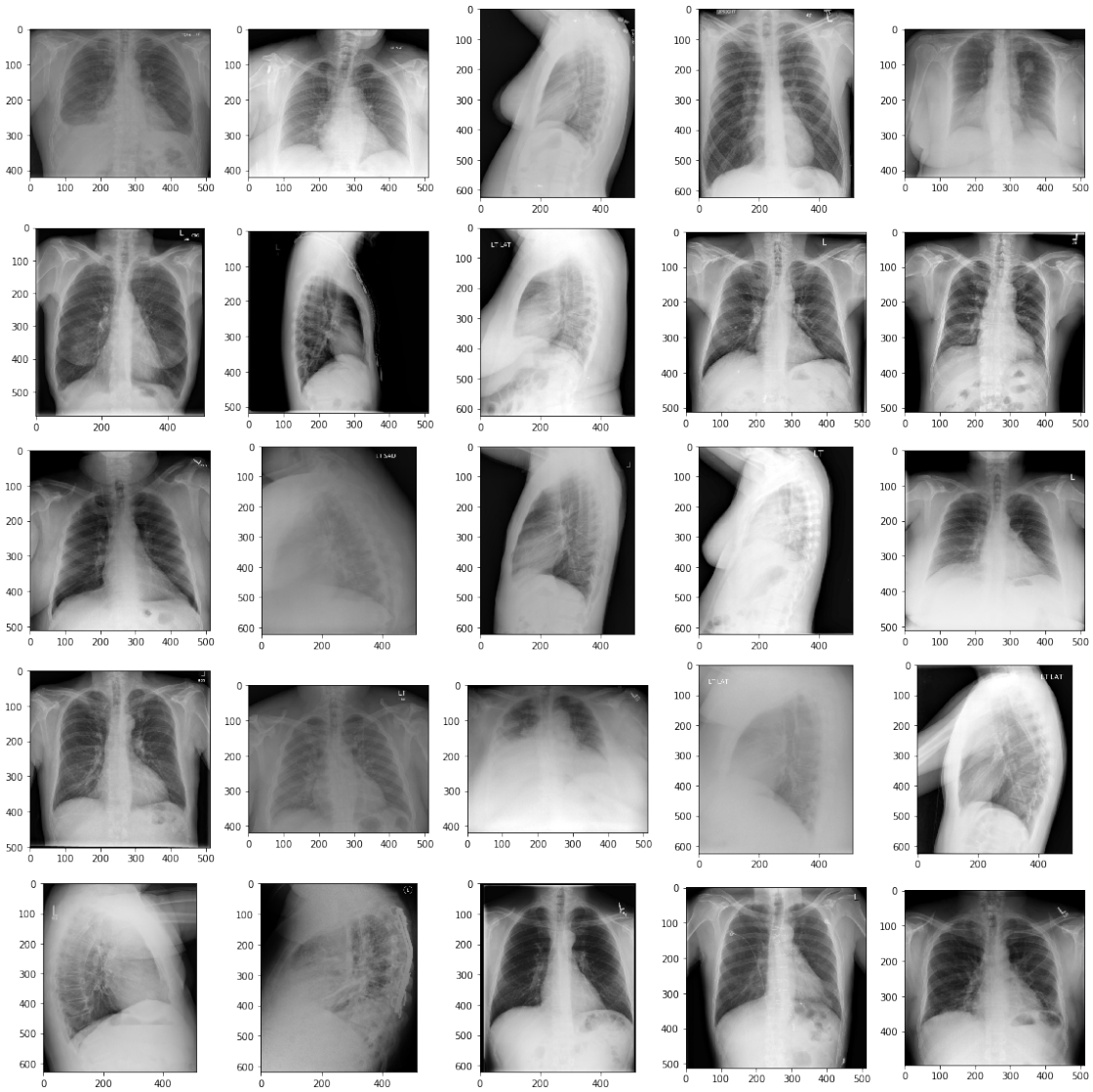
```
==== Displaying best result random 25 patient X-Ray 1st image ====
```



Points to take in best result images - proper alignment of images - brighter view of chest bones - does not have any additional dark line - Even in the dull images we could clearly see the chest bones

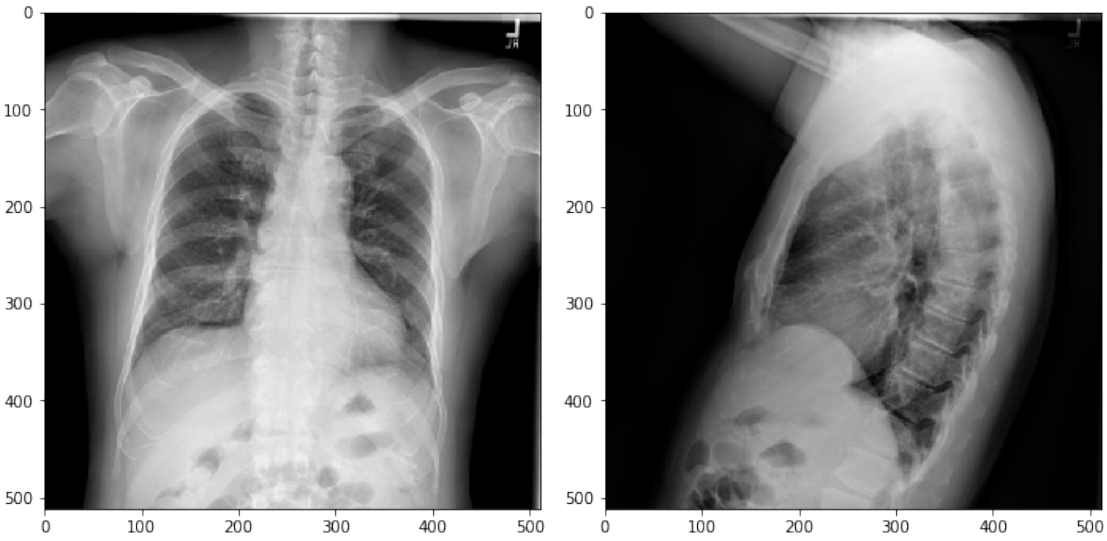
[176]:
```
print("==== Displaying worst case result random 25 patient X-Ray 1st image␣
 ↪====")
fig, axs = plt.subplots(5, 5, figsize = (16,16), tight_layout=True)
for row, subplot in zip(df_poor_zero[0:25].itertuples(), axs.flatten()):
    img=mpimg.imread(image_path+row.image_1)
    subplot.imshow(img, cmap = 'bone')
plt.show()
```

```
==== Displaying worst case result random 25 patient X-Ray 1st image ====
```



points to take in wors case - images are shadowed in some case (row, column) (3,2),(3,4),(4,2),(4,3),(4,4) - images are too bright in some case (1,2),(3,4),(5,3) - Lets see both images in a data point to check whether any one have those above issue.

[193]:
```
import random
k=random.choice(df_poor_zero.index.tolist())
img_nms = [df_poor_zero["image_1"][k], df_poor_zero["image_2"][k]]
fig, axs = plt.subplots(1, len(img_nms), figsize = (10,10), tight_layout=True)
count = 0
for img, subplot in zip(img_nms, axs.flatten()):
    img_=mpimg.imread(image_path+img)
    imgplot = axs[count].imshow(img_, cmap = 'bone')
    count +=1
plt.show()
print("Score:", df_poor_zero["score"][k])
print("Actual:", df_poor_zero["actual"][k])
print("Predicted:", df_poor_zero["predicted"][k])
print("word count:", df_poor_zero["actual_count"][k])
```




```
Score: 0.0
Actual: no evidence of active disease
Predicted: low lung features
word count: 5
```

- In this data point we see the second image is not properly taken. there is finger prints visible in the bottom of the picture, major error.

[68]:
```
k=random.choice(df_poor_zero.index.tolist())
img_nms = [df_poor_zero["image_1"][k], df_poor_zero["image_2"][k]]
fig, axs = plt.subplots(1, len(img_nms), figsize = (10,10), tight_layout=True)
count = 0
for img, subplot in zip(img_nms, axs.flatten()):
    img_=mpimg.imread(image_path+img)
```

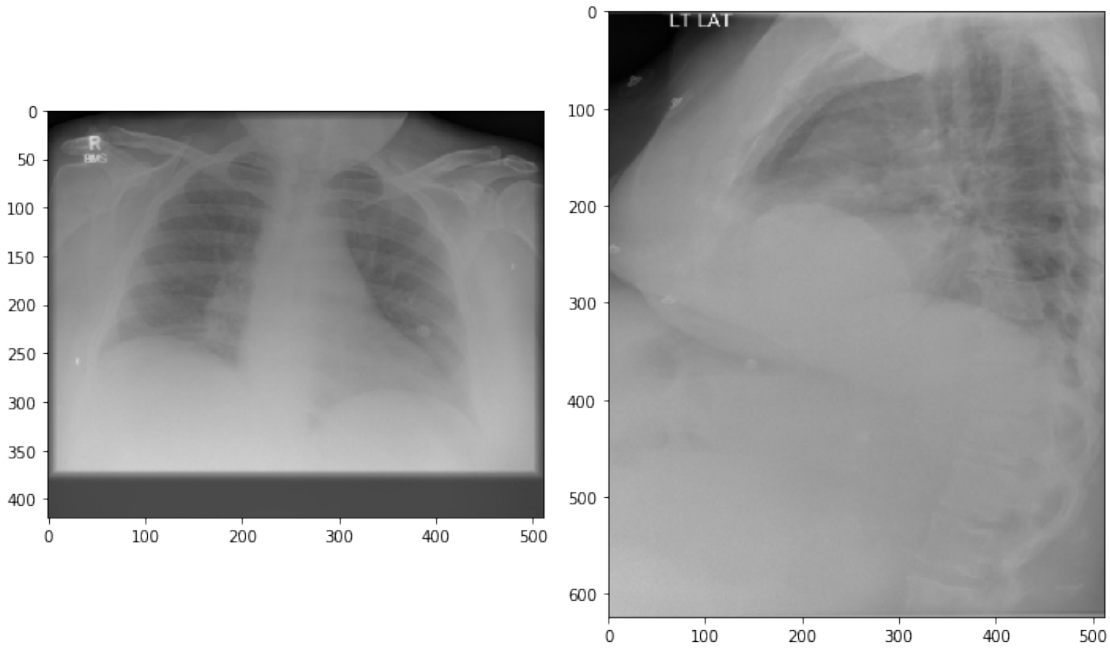
```
    imgplot = axs[count].imshow(img_, cmap = 'bone')
    count +=1
plt.show()
print("Score: ", df_poor_zero["score"][k])
print("Actual: ", df_poor_zero["actual"][k])
print("Predicted: ", df_poor_zero["predicted"][k])
```

```
Score:  0.0
Actual:  bilateral small pleural effusions and associated atelectasis stable
right upper mediastinal opacity consistent with goiter
Predicted:  no acute cardiopulmonary abnormality
```

- Clear view of poor xray capturing also covered the hands
- right side image have addition dark stripes in the lower left edge

[103]:
```
k=random.choice(df_poor_zero.index.tolist())
img_nms = [df_poor_zero["image_1"][k], df_poor_zero["image_2"][k]]
fig, axs = plt.subplots(1, len(img_nms), figsize = (10,10), tight_layout=True)
count = 0
for img, subplot in zip(img_nms, axs.flatten()):
    img_=mpimg.imread(image_path+img)
    imgplot = axs[count].imshow(img_, cmap = 'bone')
    count +=1
plt.show()
print("Score: ", df_poor_zero["score"][k])
print("Actual: ", df_poor_zero["actual"][k])
```

```
print("Predicted: ", df_poor_zero["predicted"][k])
```



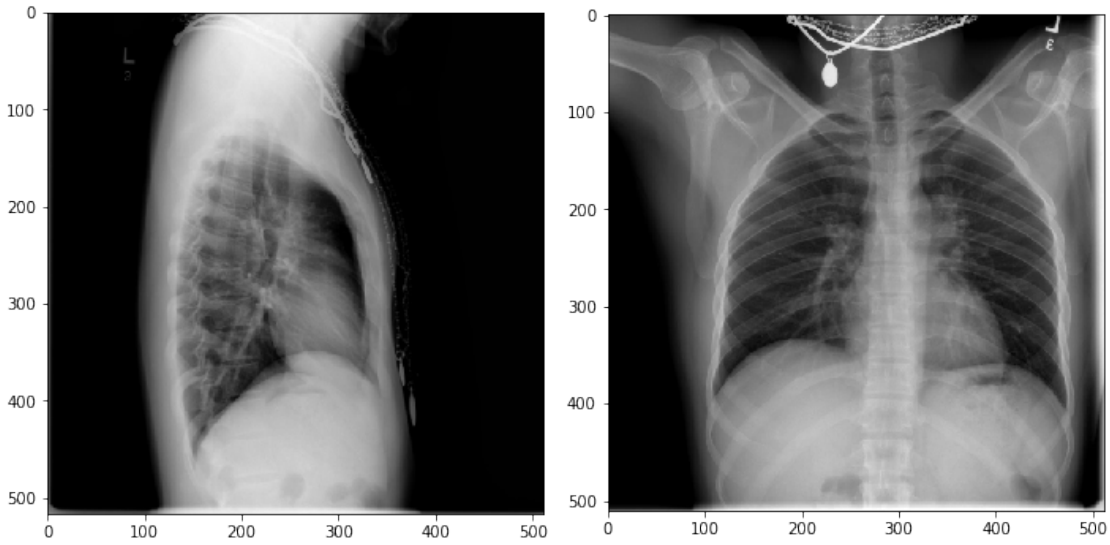

```
Score:  0.0
Actual:  mild central vascular prominence congestion heart size at the upper
limits of normal
Predicted:  no acute cardiopulmonary abnormality
```

[29]:
```
k=random.choice(df_poor_zero.index.tolist())
img_nms = [df_poor_zero["image_1"][k], df_poor_zero["image_2"][k]]
fig, axs = plt.subplots(1, len(img_nms), figsize = (10,10), tight_layout=True)
count = 0
for img, subplot in zip(img_nms, axs.flatten()):
    img_=mpimg.imread(image_path+img)
    imgplot = axs[count].imshow(img_, cmap = 'bone')
    count +=1
plt.show()
print("Score: ", df_poor_zero["score"][k])
print("Actual: ", df_poor_zero["actual"][k])
print("Predicted: ", df_poor_zero["predicted"][k])
```

```
Score:  0.0
Actual:  normal chest
Predicted:  no acute findings
```

- could not find any image quality issue. but the word meanings are same.

[26]:
```
k=random.choice(df_poor_zero.index.tolist())
img_nms = [df_poor_zero["image_1"][k], df_poor_zero["image_2"][k]]
fig, axs = plt.subplots(1, len(img_nms), figsize = (10,10), tight_layout=True)
count = 0
for img, subplot in zip(img_nms, axs.flatten()):
    img_=mpimg.imread(image_path+img)
    imgplot = axs[count].imshow(img_, cmap = 'bone')
    count +=1
plt.show()
print("Score: ", df_poor_zero["score"][k])
print("Actual: ", df_poor_zero["actual"][k])
print("Predicted: ", df_poor_zero["predicted"][k])
```



```
Score:  0.0
Actual:  no acute findings
Predicted:  low lung effusion
```

- clear view of poor image quality. x-ray with Jewellery in both images this is not found in any x-rays even in the x-ray classification task dataset.

1 Conclusion

- From the above analysis we have found that the quality of the images is plays major role. Mostly the error data points are with poor images quality poor chest bone view this is the primary take away.
- we have also seen some finger prints, jewellery of the patient clearly visible in the image.
- the model works well on the clear visible chest bones. we have already seen this and compared in the best and worst case images.
- there are images which are brighter those cases model fails. we have also seen the best result images where we does not have the brighter images. brighter means higher white pixels.
- Another finding is that our model did not perform well in the case where we have more than 20 words. we could able to improve this by changing the architecture. better than this but our model does not shows that its poor model. error are 62 out of 399 which is almost 15% of the data. does not show it is poor prediction.
- some case where we have incorrect words in true sentence.
- we could ignore theses error in our future work to get the better performance.

[ ]: