

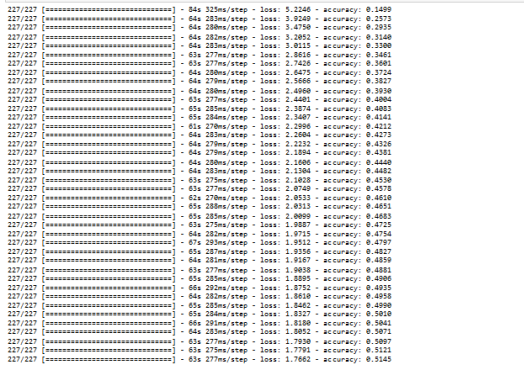

Project Development Phase

Model Performance Test

Date	22 November 2023
Team ID	591718
Project Name	Image Caption Generation
Maximum Marks	10 Marks

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

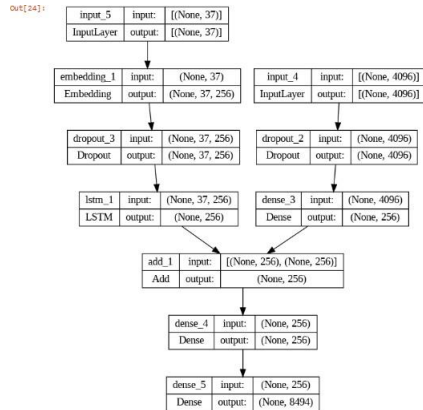
S.No.	Parameter	Values	Screenshot
1.	Metrics	Regression Model: BLUE score, Accuracy and Loss.	<pre>In []: # train the model epochs = 40 batch_size = 32 steps = len(train) // batch_size for i in range(epochs): # create data generator generator = data_generator(train, mapping, features, tokenizer, max_length, vocab_size, batch_size) # fit for one epoch model.fit(generator, epochs=1, steps_per_epoch=steps, verbose=1.)</pre>  <pre>227/227 [#####] - 04s 320ms/step - loss: 5.2246 - accuracy: 0.1499 227/227 [#####] - 04s 280ms/step - loss: 3.9240 - accuracy: 0.2373 227/227 [#####] - 04s 280ms/step - loss: 3.4780 - accuracy: 0.3035 227/227 [#####] - 04s 280ms/step - loss: 3.2852 - accuracy: 0.3148 227/227 [#####] - 04s 280ms/step - loss: 3.0515 - accuracy: 0.3388 227/227 [#####] - 03s 277ms/step - loss: 2.8616 - accuracy: 0.3461 227/227 [#####] - 03s 277ms/step - loss: 2.7428 - accuracy: 0.3601 227/227 [#####] - 04s 280ms/step - loss: 2.6475 - accuracy: 0.3724 227/227 [#####] - 04s 279ms/step - loss: 2.5665 - accuracy: 0.3827 227/227 [#####] - 04s 280ms/step - loss: 2.4868 - accuracy: 0.3930 227/227 [#####] - 03s 277ms/step - loss: 2.4401 - accuracy: 0.4084 227/227 [#####] - 03s 280ms/step - loss: 2.3874 - accuracy: 0.4082 227/227 [#####] - 03s 284ms/step - loss: 2.3487 - accuracy: 0.4141 227/227 [#####] - 03s 279ms/step - loss: 2.2986 - accuracy: 0.4112 227/227 [#####] - 04s 283ms/step - loss: 2.2684 - accuracy: 0.4273 227/227 [#####] - 04s 279ms/step - loss: 2.2332 - accuracy: 0.4326 227/227 [#####] - 04s 279ms/step - loss: 2.1894 - accuracy: 0.4381 227/227 [#####] - 04s 280ms/step - loss: 2.1606 - accuracy: 0.4440 227/227 [#####] - 04s 283ms/step - loss: 2.1384 - accuracy: 0.4482 227/227 [#####] - 03s 277ms/step - loss: 2.1823 - accuracy: 0.4538 227/227 [#####] - 03s 277ms/step - loss: 2.0769 - accuracy: 0.4575 227/227 [#####] - 03s 279ms/step - loss: 2.0533 - accuracy: 0.4610 227/227 [#####] - 03s 280ms/step - loss: 2.0313 - accuracy: 0.4651 227/227 [#####] - 03s 289ms/step - loss: 2.0089 - accuracy: 0.4683 227/227 [#####] - 03s 279ms/step - loss: 1.9807 - accuracy: 0.4725 227/227 [#####] - 04s 282ms/step - loss: 1.9715 - accuracy: 0.4754 227/227 [#####] - 07s 295ms/step - loss: 1.9512 - accuracy: 0.4797 227/227 [#####] - 03s 287ms/step - loss: 1.9356 - accuracy: 0.4827 227/227 [#####] - 04s 285ms/step - loss: 1.9167 - accuracy: 0.4839 227/227 [#####] - 03s 277ms/step - loss: 1.9488 - accuracy: 0.4851 227/227 [#####] - 03s 289ms/step - loss: 1.8995 - accuracy: 0.4906 227/227 [#####] - 04s 292ms/step - loss: 1.8752 - accuracy: 0.4935 227/227 [#####] - 04s 282ms/step - loss: 1.8648 - accuracy: 0.4950 227/227 [#####] - 03s 280ms/step - loss: 1.8462 - accuracy: 0.4998 227/227 [#####] - 03s 284ms/step - loss: 1.8327 - accuracy: 0.5010 227/227 [#####] - 06s 295ms/step - loss: 1.8188 - accuracy: 0.5041 227/227 [#####] - 04s 283ms/step - loss: 1.8082 - accuracy: 0.5071 227/227 [#####] - 03s 277ms/step - loss: 1.7938 - accuracy: 0.5097 227/227 [#####] - 03s 279ms/step - loss: 1.7781 - accuracy: 0.5121 227/227 [#####] - 03s 277ms/step - loss: 1.7662 - accuracy: 0.5145</pre>
		BLUE SCORE-	<pre>In []: from itertools import zip_longest import corpus_bleu # validate with test data actual, predicted = list(), list() for key in tqdm(test): # get actual caption captions = mapping[key] # predict the caption for image y_pred = predict_caption(model, features[key], tokenizer, max_length) # split into words actual_captions = [caption.split() for caption in captions] y_pred = y_pred.split() # append to the list actual.append(actual_captions) predicted.append(y_pred) # calculate BLEU score print("BLEU-1: %f" % corpus_bleu(actual, predicted, weights=(1,0,0,0,0))) print("BLEU-2: %f" % corpus_bleu(actual, predicted, weights=(0.5,0.5,0,0,0)))</pre>  <pre>BLEU-1: 0.322298 BLEU-1: 0.537946 BLEU-1: 0.322338 BLEU-1: 0.537433 BLEU-1: 0.322386 BLEU-1: 0.537946 BLEU-1: 0.322348 BLEU-1: 0.537993 BLEU-1: 0.322778 BLEU-1: 0.536933 BLEU-1: 0.322662 BLEU-1: 0.536987 BLEU-1: 0.322554 BLEU-1: 0.537044 BLEU-1: 0.322544 BLEU-1: 0.537046 BLEU-1: 0.322559 BLEU-1: 0.537772 BLEU-1: 0.322390</pre>

2. Tune the Model

Hyperparameter Tuning - Validation Method -

Hyperparamter tuning

```
In [ ]: # encoder model
# large feature layers
inputs1 = Input(shape=(699,))
fe1 = Dropout(0.5)(inputs1)
fe2 = Dense(256, activation='relu')(fe1)
# sequence feature layers
inputs2 = Input(shape=(max_length,))
se1 = Embedding(vocab_size, 256, mask_zero=True)(inputs2)
se2 = Dropout(0.5)(se1)
se3 = LSTM(256)(se2)
# decoder model
decoder1 = add([fe2, se3])
decoder2 = Dense(256, activation='relu')(decoder1)
outputs = Dense(vocab_size, activation='softmax')(decoder2)
model = Model(inputs=[inputs1, inputs2], outputs=outputs)
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
# plot the model
plot_model(model, show_shapes=True)
```



Validation-

```
In [ ]: generate_caption("0801773457_977c3a7d7b.jpg")
```

-----Actual-----
start black dog and spotted dog are fighting and
start black dog and tri-colored dog playing with each other on the road and
start black dog and white dog with brown spots are staring at each other in the street and
start two dogs of different breeds looking at each other on the road and
start two dogs on pavement moving toward each other and
-----Predicted-----
start black dog and spotted dog playing on pavement and



```
In [ ]: generate_caption("0802074543_1b742ab40b.jpg")
```

-----Actual-----
start little girl covered in paint sits in front of painted rainbow with her hands in bowl and
start little girl is sitting in front of large painted rainbow and
start small girl in the grass plays with fingerpaints in front of white canvas with rainbow on it and
start there is girl with pigtails sitting in front of rainbow painting and
start young girl with pigtails painting outside in the grass and
-----Predicted-----
start two children playing in plastic colored plastic structure and



```

In [ ]: generate_caption("305689240_b059c7f17b.jpg")

-----Actual-----
start man in hat is displaying pictures next to skier in blue hat and
start man skis past another man displaying paintings in the snow end
start person wearing skis looking at framed pictures set up in the snow end
start skier looks at framed pictures in the snow next to trees and
start man on skis looking at artwork for sale in the snow end
-----Predicted-----
start man in skis is displaying paintings in the snow end

```

