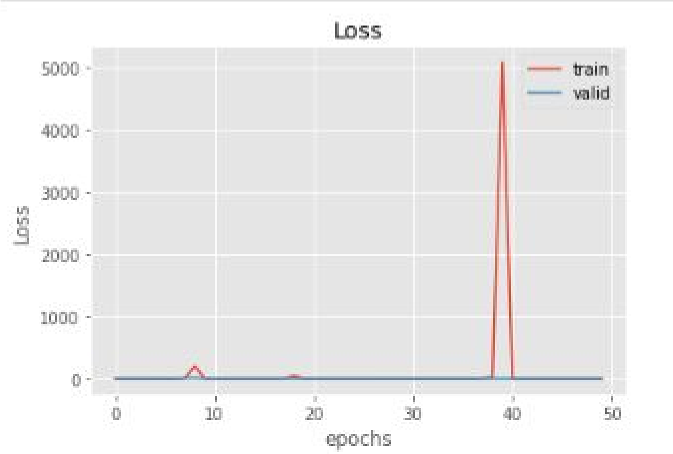
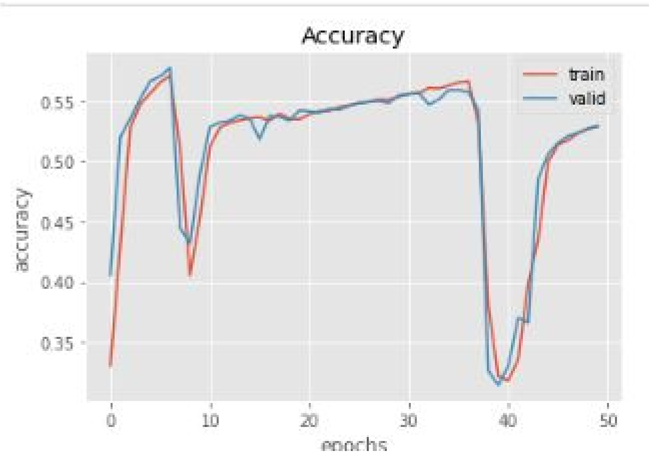
Experimentos Iniciais

```
#b2wCorpusTrain=b2wCorpusTrain.reindex(b2wCorpusTrain['ord'].sort_values(ascending=False).index)

from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(layers.LSTM(128, activation='relu',return_sequences=True))
#model.add(Dropout(0.2))
model.add(keras.layers.LSTM(128, activation='relu'))
#model.add(Dropout(0.2))
model.add(keras.layers.Dense(32, activation='relu'))
#model.add(Dropout(0.2))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
    x= x_train, y=y_train, batch_size=128, epochs=50, validation_data=(x_val, y_val)
)
```





```
history = model.fit(
    x= x_train, y=y_train, batch_size=32, epochs=50, validation_data=(x_val, y_val)
)
```

```
Epoch 16/50
1390/1390 [==============================] - 63s 46ms/step - loss: 5.2713 - accuracy: 0.4451 - val_loss: 2.4653 -
val_accuracy: 0.3456
Epoch 17/50
1390/1390 [==============================] - 63s 46ms/step - loss: 3.9887 - accuracy: 0.4126 - val_loss: 15.7436 -
val_accuracy: 0.4289
Epoch 18/50
1390/1390 [==============================] - 62s 45ms/step - loss: 2.8532 - accuracy: 0.4621 - val_loss: 1.2931 -
val_accuracy: 0.4956
Epoch 19/50
1390/1390 [==============================] - 56s 40ms/step - loss: 1.9064 - accuracy: 0.4979 - val_loss: 1.2428 -
val_accuracy: 0.5047
Epoch 20/50
1390/1390 [==============================] - 65s 47ms/step - loss: 1.2510 - accuracy: 0.5116 - val_loss: 1.1823 -
val_accuracy: 0.5199
Epoch 21/50
1390/1390 [==============================] - 62s 45ms/step - loss: 1.1641 - accuracy: 0.5270 - val_loss: 1.1489 -
val_accuracy: 0.5304
Epoch 22/50
1390/1390 [==============================] - 63s 45ms/step - loss: 1.1359 - accuracy: 0.5345 - val_loss: 1.1366 -
val_accuracy: 0.5327
Epoch 23/50
1390/1390 [==============================] - 57s 41ms/step - loss: 1.1199 - accuracy: 0.5393 - val_loss: 1.1196 -
val_accuracy: 0.5357
Epoch 24/50
1390/1390 [==============================] - 55s 40ms/step - loss: 1.1070 - accuracy: 0.5421 - val_loss: 1.1081 -
val_accuracy: 0.5395
Epoch 25/50
1390/1390 [==============================] - 63s 45ms/step - loss: 1.1000 - accuracy: 0.5418 - val_loss: 1.1108 -
val_accuracy: 0.5374
Epoch 26/50
1390/1390 [==============================] - 64s 46ms/step - loss: 1.0895 - accuracy: 0.5457 - val_loss: 1.0945 -
val_accuracy: 0.5456
Epoch 27/50
1390/1390 [==============================] - 64s 46ms/step - loss: 1.0683 - accuracy: 0.5532 - val_loss: 1.0771 -
val_accuracy: 0.5414
Epoch 28/50
```

```
1390/1390 [==============================] - 63s 45ms/step - loss: 7678.0850 - accuracy: 0.3338 - val_loss: 1.5071 -
val_accuracy: 0.3174
Epoch 29/50
1390/1390 [==============================] - 60s 43ms/step - loss: 38.4642 - accuracy: 0.3197 - val_loss: 1.7101 -
val_accuracy: 0.3368
```

```
history = model.fit(
    x= x_train, y=y_train, batch_size=1024, epochs=50, validation_data=(x_val, y_val)
)
```

```
Epoch 1/50
44/44 [==============================] - 20s 451ms/step - loss: 1.5287 - accuracy: 0.3243 - val_loss: 1.4915
- val_accuracy: 0.3601
Epoch 2/50
44/44 [==============================] - 19s 443ms/step - loss: 1.4437 - accuracy: 0.3964 - val_loss: 1.2711
- val_accuracy: 0.4966
Epoch 3/50
44/44 [==============================] - 19s 432ms/step - loss: 1.2456 - accuracy: 0.5066 - val_loss: 1.2191
- val_accuracy: 0.5152
Epoch 4/50
44/44 [==============================] - 19s 432ms/step - loss: 3.9013 - accuracy: 0.3587 - val_loss: 1.5117
- val_accuracy: 0.3292
Epoch 5/50
44/44 [==============================] - 19s 432ms/step - loss: 1.4809 - accuracy: 0.3482 - val_loss: 1.3612
- val_accuracy: 0.4570
Epoch 6/50
44/44 [==============================] - 19s 440ms/step - loss: 1.3022 - accuracy: 0.5000 - val_loss: 1.2242
- val_accuracy: 0.5130
Epoch 7/50
31/44 [====================>.........] - ETA: 5s - loss: 1.2017 - accuracy: 0.5160
```

```
history = model.fit(
    x= x_train, y=y_train, batch_size=256, epochs=50, validation_data=(x_val, y_val)
)
```

```
Epoch 1/50
174/174 [==============================] - 24s 137ms/step - loss: 1.3727 - accuracy: 0.4368 - val_loss:
1.2105 - val_accuracy: 0.5206
Epoch 2/50
174/174 [==============================] - 24s 136ms/step - loss: 1.2879 - accuracy: 0.5286 - val_loss:
1.1205 - val_accuracy: 0.5406
Epoch 3/50
174/174 [==============================] - 24s 140ms/step - loss: 1.0950 - accuracy: 0.5462 - val_loss:
1.0877 - val_accuracy: 0.5468
Epoch 4/50
174/174 [==============================] - 24s 140ms/step - loss: 1.0586 - accuracy: 0.5562 - val_loss:
1.0499 - val_accuracy: 0.5534
Epoch 5/50
174/174 [==============================] - 24s 140ms/step - loss: 1.0688 - accuracy: 0.5570 - val_loss:
1.0577 - val_accuracy: 0.5563
Epoch 6/50
174/174 [==============================] - 24s 140ms/step - loss: 1.0297 - accuracy: 0.5666 - val_loss:
1.0170 - val_accuracy: 0.5665
```

```
Epoch 7/50
174/174 [==============================] - 24s 140ms/step - loss: 1.0099 - accuracy: 0.5741 - val_loss:
1.0052 - val_accuracy: 0.5737
Epoch 8/50
174/174 [==============================] - 24s 140ms/step - loss: 0.9908 - accuracy: 0.5832 - val_loss:
1.0105 - val_accuracy: 0.5756
Epoch 9/50
174/174 [==============================] - 24s 138ms/step - loss: 22.8742 - accuracy: 0.4128 - val_loss:
1.7813 - val_accuracy: 0.3073
Epoch 10/50
174/174 [==============================] - 25s 143ms/step - loss: 1.5137 - accuracy: 0.3297 - val_loss:
1.5028 - val_accuracy: 0.3405
Epoch 11/50
174/174 [==============================] - 24s 139ms/step - loss: 1.4968 - accuracy: 0.3590 - val_loss:
1.7890 - val_accuracy: 0.4077
Epoch 12/50
174/174 [==============================] - 28s 163ms/step - loss: 1.5475 - accuracy: 0.3848 - val_loss:
1.4806 - val_accuracy: 0.3689
```

```
history = model.fit(
   x= x_train, y=y_train, batch_size=128, epochs=50, validation_data=(x_val, y_val)
)
```

```
Epoch 1/50
348/348 [==============================] - 29s 85ms/step - loss: 7.4891 - accuracy: 0.3805 - val_loss: 15.9545 -
val_accuracy: 0.5113
Epoch 2/50
348/348 [==============================] - 29s 84ms/step - loss: 2.9377 - accuracy: 0.4564 - val_loss: 1.5198 -
val_accuracy: 0.3123
Epoch 3/50
348/348 [==============================] - 29s 84ms/step - loss: 1.5113 - accuracy: 0.3273 - val_loss: 1.4961 -
val_accuracy: 0.3409
Epoch 4/50
348/348 [==============================] - 29s 83ms/step - loss: 1.5084 - accuracy: 0.3530 - val_loss: 1.4954 -
val_accuracy: 0.3621
Epoch 5/50
348/348 [==============================] - 29s 84ms/step - loss: 1.3743 - accuracy: 0.4388 - val_loss: 1.1830 -
val_accuracy: 0.5234
Epoch 6/50
348/348 [==============================] - 30s 86ms/step - loss: 1.1621 - accuracy: 0.5277 - val_loss: 1.1199 -
val_accuracy: 0.5379
Epoch 7/50
348/348 [==============================] - 30s 87ms/step - loss: 1.1229 - accuracy: 0.5378 - val_loss: 1.1889 -
val_accuracy: 0.5380
Epoch 8/50
348/348 [==============================] - 30s 86ms/step - loss: 1.1119 - accuracy: 0.5407 - val_loss: 1.0996 -
val_accuracy: 0.5437
Epoch 9/50
348/348 [==============================] - 29s 84ms/step - loss: 1.7999 - accuracy: 0.4420 - val_loss: 1.3428 -
val_accuracy: 0.4611
Epoch 10/50
348/348 [==============================] - 29s 84ms/step - loss: 1.2746 - accuracy: 0.4897 - val_loss: 1.2243 -
val_accuracy: 0.5056
Epoch 11/50
348/348 [==============================] - 30s 87ms/step - loss: 1.2189 - accuracy: 0.5095 - val_loss: 1.1985 -
val_accuracy: 0.5158
Epoch 12/50
```

```
348/348 [==============================] - 30s 86ms/step - loss: 782607.9375 - accuracy: 0.5160 - val_loss: 2.4031 -
val_accuracy: 0.2798
Epoch 13/50
348/348 [==============================] - 32s 92ms/step - loss: 2225291.0000 - accuracy: 0.3117 - val_loss: 4.4533 -
val_accuracy: 0.3890
```

```
from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(layers.LSTM(256, activation='relu',return_sequences=True))
model.add(Dropout(0.2))
model.add(keras.layers.LSTM(128, activation='relu'))
model.add(Dropout(0.2))
model.add(keras.layers.Dense(64, activation='relu'))
model.add(Dropout(0.2))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
   x= x_train, y=y_train, batch_size=128, epochs=50, validation_data=(x_val, y_val)
```

```
Epoch 1/50
348/348 [==============================] - 71s 203ms/step - loss: 8.2440 - accuracy: 0.3096 -
val_loss: 1.5044 - val_accuracy: 0.3135
Epoch 2/50
348/348 [==============================] - 71s 204ms/step - loss: 375218700288.0000 - accuracy:
0.2987 - val_loss: 45062.9492 - val_accuracy: 0.1994
Epoch 3/50
131/348 [==========>...................] - ETA: 42s - loss: 124356.7969 - accuracy: 0.2287
```

```
from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(layers.LSTM(256, activation='relu',return_sequences=True))
model.add(Dropout(0.5))
model.add(keras.layers.LSTM(128, activation='relu'))
model.add(Dropout(0.5))
model.add(keras.layers.Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
    x= x_train, y=y_train, batch_size=128, epochs=50, validation_data=(x_val, y_val)
```

```
Epoch 1/50
348/348 [==============================] - 71s 203ms/step - loss: 1.5241 - accuracy: 0.3278 -
val_loss: 1.5046 - val_accuracy: 0.3161
Epoch 2/50
348/348 [==============================] - 67s 194ms/step - loss: 1.4668 - accuracy: 0.3827 -
val_loss: 1.2288 - val_accuracy: 0.5231
Epoch 3/50
348/348 [==============================] - 64s 185ms/step - loss: 2.2067 - accuracy: 0.5123 -
val_loss: 6.2918 - val_accuracy: 0.2844
Epoch 4/50
348/348 [==============================] - 56s 161ms/step - loss: 15.1495 - accuracy: 0.4409 -
val_loss: 1.4869 - val_accuracy: 0.3586
Epoch 5/50
 83/348 [======>.......................] - ETA: 44s - loss: 1.4693 - accuracy: 0.3944
```

```
from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(layers.LSTM(256, activation='relu'))
model.add(Dropout(0.5))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
    x= x_train, y=y_train, batch_size=128, epochs=50, validation_data=(x_val, y_val)
```

```
Epoch 1/50
```

```
348/348 [==============================] - 36s 103ms/step - loss: 24.6264 - accuracy: 0.3093 - val_loss:
1.5049 - val_accuracy: 0.3200
Epoch 2/50
348/348 [==============================] - 36s 102ms/step - loss: 1.5065 - accuracy: 0.3136 - val_loss:
3.5476 - val_accuracy: 0.3279
Epoch 3/50
348/348 [==============================] - 38s 109ms/step - loss: 1.5278 - accuracy: 0.3063 - val_loss:
1.5012 - val_accuracy: 0.3256
Epoch 4/50
348/348 [==============================] - 42s 121ms/step - loss: 1.5100 - accuracy: 0.3139 - val_loss:
1.5029 - val_accuracy: 0.3189
Epoch 5/50
348/348 [==============================] - 41s 119ms/step - loss: 1.5106 - accuracy: 0.3173 - val_loss:
1.5033 - val_accuracy: 0.3250
Epoch 6/50
249/348 [====================>.........] - ETA: 11s - loss: 1.5071 - accuracy: 0.3299
```

```
from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(layers.LSTM(256, activation='relu'))
model.add(Dropout(0.5))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
   x= x_train, y=y_train, batch_size=32, epochs=50, validation_data=(x_val, y_val)
)
```

```
Epoch 1/50
1390/1390 [==============================] - 72s 52ms/step - loss: 1.6533 - accuracy: 0.3074 - val_loss:
1.5047 - val_accuracy: 0.3472
Epoch 2/50
1390/1390 [==============================] - 66s 47ms/step - loss: 1.5113 - accuracy: 0.3099 - val_loss:
1.5030 - val_accuracy: 0.3194
Epoch 3/50
1390/1390 [==============================] - 69s 50ms/step - loss: 1.5090 - accuracy: 0.3169 - val_loss:
1.5023 - val_accuracy: 0.3228
Epoch 4/50
 504/1390 [=========>....................] - ETA: 42s - loss: 1.5049 - accuracy: 0.3220
```

```
from tensorflow.keras.losses import sparse_categorical_crossentropy
```

```
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(layers.LSTM(64, activation='relu'))
model.add(Dropout(0.5))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
    x= x_train, y=y_train, batch_size=32, epochs=50, validation_data=(x_val, y_val)
```

```
Epoch 1/50
1390/1390 [==============================] - 22s 16ms/step - loss: 1.5228 - accuracy: 0.3155 -
val_loss: 1.4873 - val_accuracy: 0.3639
Epoch 2/50
1390/1390 [==============================] - 21s 15ms/step - loss: 1.5138 - accuracy: 0.3126 -
val_loss: 1.5024 - val_accuracy: 0.3235
Epoch 3/50
1390/1390 [==============================] - 21s 15ms/step - loss: 1.5050 - accuracy: 0.3328 -
val_loss: 1.4949 - val_accuracy: 0.3395
Epoch 4/50
1390/1390 [==============================] - 20s 15ms/step - loss: 3.1076 - accuracy: 0.3582 -
val_loss: 2.0512 - val_accuracy: 0.3770
Epoch 5/50
1390/1390 [==============================] - 20s 15ms/step - loss: 397.6962 - accuracy: 0.3562 -
val_loss: 1.4617 - val_accuracy: 0.4439
Epoch 6/50
1390/1390 [==============================] - 20s 15ms/step - loss: 1.3790 - accuracy: 0.4647 -
val_loss: 1.3039 - val_accuracy: 0.4920
Epoch 7/50
1390/1390 [==============================] - 20s 15ms/step - loss: 1.3348 - accuracy: 0.5038 -
val_loss: 1.6243 - val_accuracy: 0.5018
Epoch 8/50
1390/1390 [==============================] - 20s 15ms/step - loss: 1.4248 - accuracy: 0.5022 -
val_loss: 1.2491 - val_accuracy: 0.5041
Epoch 9/50
1390/1390 [==============================] - 20s 15ms/step - loss: 1.2582 - accuracy: 0.5132 -
val_loss: 1.1791 - val_accuracy: 0.5316
Epoch 10/50
1390/1390 [==============================] - 20s 15ms/step - loss: 1.1978 - accuracy: 0.5277 -
val_loss: 1.1764 - val_accuracy: 0.5338
Epoch 11/50
```

```
1390/1390 [==============================] - 21s 15ms/step - loss: 1.4174 - accuracy: 0.5239 -
val_loss: 1.1411 - val_accuracy: 0.5348
Epoch 12/50
1390/1390 [==============================] - 22s 16ms/step - loss: 1.1624 - accuracy: 0.5333 -
val_loss: 1.1181 - val_accuracy: 0.5406
Epoch 13/50
1390/1390 [==============================] - 22s 16ms/step - loss: 1.1732 - accuracy: 0.5309 -
val_loss: 1.1254 - val_accuracy: 0.5398
Epoch 14/50
1390/1390 [==============================] - 22s 16ms/step - loss: 1.1548 - accuracy: 0.5363 -
val_loss: 2.2781 - val_accuracy: 0.2640
Epoch 15/50
1390/1390 [==============================] - 22s 16ms/step - loss: 12.3980 - accuracy: 0.3341 -
val_loss: 2.5758 - val_accuracy: 0.3456
Epoch 16/50
1390/1390 [==============================] - 22s 16ms/step - loss: 2.4581 - accuracy: 0.3381 -
val_loss: 27.2640 - val_accuracy: 0.2836
Epoch 17/50
1390/1390 [==============================] - 22s 16ms/step - loss: 6.4993 - accuracy: 0.3374 -
val_loss: 1.6702 - val_accuracy: 0.4197
Epoch 18/50
1390/1390 [==============================] - 22s 16ms/step - loss: 1.7337 - accuracy: 0.3458 -
val_loss: 1.5727 - val_accuracy: 0.4515
Epoch 19/50
1390/1390 [==============================] - 22s 16ms/step - loss: 1.6160 - accuracy: 0.3714 -
val_loss: 1.4636 - val_accuracy: 0.4670
Epoch 20/50
1390/1390 [==============================] - 22s 16ms/step - loss: 1.4923 - accuracy: 0.4225 -
val_loss: 1.4065 - val_accuracy: 0.4747
Epoch 21/50
1390/1390 [==============================] - 22s 16ms/step - loss: 1.4321 - accuracy: 0.4529 -
val_loss: 1.3354 - val_accuracy: 0.4892
Epoch 22/50
```

```
rom tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(layers.LSTM(64, activation='relu'))
#model.add(Dropout(0.5))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
```

```
history = model.fit(
   x= x_train, y=y_train, batch_size=32, epochs=50, validation_data=(x_val, y_val)
)

Epoch 1/50
1390/1390 [==============================] - 23s 17ms/step - loss: 1.5401 - accuracy:
0.3159 - val_loss: 1.4995 - val_accuracy: 0.3275
Epoch 2/50
1390/1390 [==============================] - 22s 16ms/step - loss: 1.5111 - accuracy:
0.3277 - val_loss: 1.5008 - val_accuracy: 0.3386
Epoch 3/50
1390/1390 [==============================] - 22s 16ms/step - loss: 1.5352 - accuracy:
0.3454 - val_loss: 1.5667 - val_accuracy: 0.2841
Epoch 4/50
1390/1390 [==============================] - 22s 16ms/step - loss: 1.7110 - accuracy:
0.3534 - val_loss: 1.4353 - val_accuracy: 0.4177
Epoch 5/50
1390/1390 [==============================] - 22s 15ms/step - loss: 1.7573 - accuracy:
0.4266 - val_loss: 1.2817 - val_accuracy: 0.4933
Epoch 6/50
1390/1390 [==============================] - 21s 15ms/step - loss: 1.2286 - accuracy:
0.5093 - val_loss: 1.1784 - val_accuracy: 0.5216
Epoch 7/50
1390/1390 [==============================] - 21s 15ms/step - loss: 1.1709 - accuracy:
0.5265 - val_loss: 1.1313 - val_accuracy: 0.5355
Epoch 8/50
1390/1390 [==============================] - 21s 15ms/step - loss: 1.1282 - accuracy:
0.5363 - val_loss: 1.1261 - val_accuracy: 0.5336
Epoch 9/50
1390/1390 [==============================] - 21s 15ms/step - loss: 1.1022 - accuracy:
0.5418 - val_loss: 1.0844 - val_accuracy: 0.5468
Epoch 10/50
1390/1390 [==============================] - 21s 15ms/step - loss: 1.0789 - accuracy:
0.5481 - val_loss: 1.0645 - val_accuracy: 0.5513
Epoch 11/50
1390/1390 [==============================] - 21s 15ms/step - loss: 1.6132 - accuracy:
0.5344 - val_loss: 1.1085 - val_accuracy: 0.5439
Epoch 12/50
1390/1390 [==============================] - 21s 15ms/step - loss: 1.0744 - accuracy:
0.5509 - val_loss: 1.0675 - val_accuracy: 0.5557
Epoch 13/50
1390/1390 [==============================] - 21s 15ms/step - loss: 1.0562 - accuracy:
0.5598 - val_loss: 1.0615 - val_accuracy: 0.5602
Epoch 14/50
```

```
1390/1390 [==============================] - 21s 15ms/step - loss: 1.0433 - accuracy:
0.5630 - val_loss: 1.0696 - val_accuracy: 0.5667
Epoch 15/50
1390/1390 [==============================] - 21s 15ms/step - loss: 1.0326 - accuracy:
0.5700 - val_loss: 1.0413 - val_accuracy: 0.5671
Epoch 16/50
1390/1390 [==============================] - 21s 15ms/step - loss: 1.0159 - accuracy:
0.5783 - val_loss: 1.0217 - val_accuracy: 0.5759
Epoch 17/50
1390/1390 [==============================] - 22s 16ms/step - loss: 1.0100 - accuracy:
0.5807 - val_loss: 1.0334 - val_accuracy: 0.5746
Epoch 18/50
1390/1390 [==============================] - 21s 15ms/step - loss: 319.0436 -
accuracy: 0.4959 - val_loss: 1.8702 - val_accuracy: 0.2792
Epoch 19/50
1390/1390 [==============================] - 22s 16ms/step - loss: 1.5047 - accuracy:
0.3812 - val_loss: 1.4549 - val_accuracy: 0.4310
Epoch 20/50
 859/1390 [=================>............] - ETA: 7s - loss: 12269.5859 - accuracy: 0.4041
```

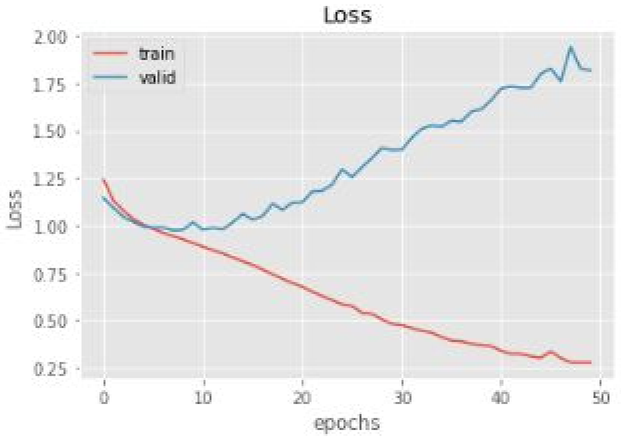
```
from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(layers.LSTM(128, activation='relu'))
#model.add(Dropout(0.5))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
    x= x_train, y=y_train, batch_size=32, epochs=50, validation_data=(x_val, y_val)
)
```

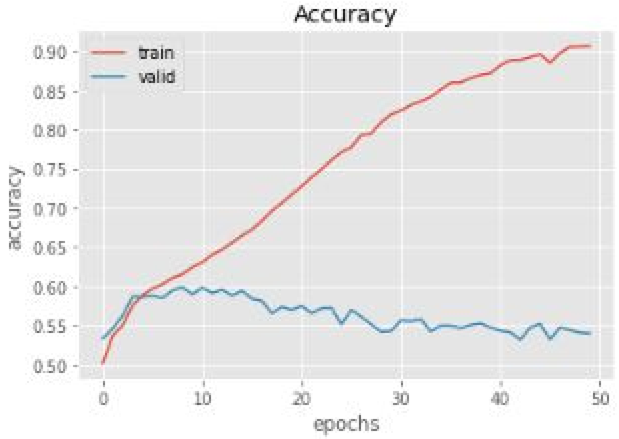
```
Epoch 1/50
1390/1390 [==============================] - 35s 25ms/step - loss: 1.5107 - accuracy: 0.3252 - val_loss:
1.4463 - val_accuracy: 0.4708
Epoch 2/50
1390/1390 [==============================] - 36s 26ms/step - loss: 1.9591 - accuracy: 0.3422 - val_loss:
1.5012 - val_accuracy: 0.3314
Epoch 3/50
```

```
1390/1390 [==============================] - 36s 26ms/step - loss: 1.5106 - accuracy: 0.3286 - val_loss:
1.5004 - val_accuracy: 0.3371
Epoch 4/50
1390/1390 [==============================] - 32s 23ms/step - loss: 1.5018 - accuracy: 0.3415 - val_loss:
1.5008 - val_accuracy: 0.3257
Epoch 5/50
 515/1390 [==========>...................] - ETA: 19s - loss: 1014.9067 - accuracy: 0.3247
```

```
from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(layers.LSTM(128))
#model.add(Dropout(0.5))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
    x= x_train, y=y_train, batch_size=32, epochs=50, validation_data=(x_val, y_val)
)
```

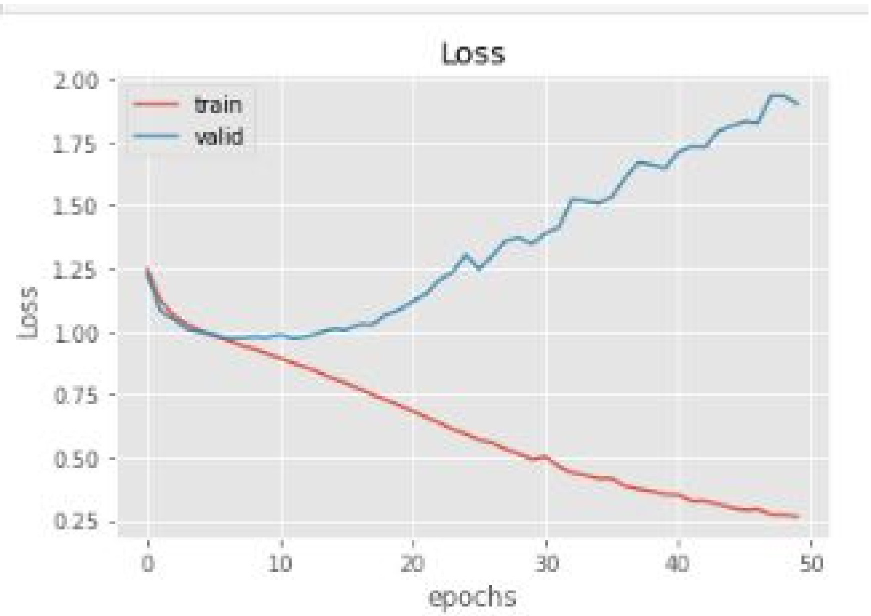
```
poch 48/50
1390/1390 [==============================] - 31s 22ms/step - loss: 0.2783 - accuracy:
0.9054 - val_loss: 1.9403 - val_accuracy: 0.5439
Epoch 49/50
1390/1390 [==============================] - 30s 22ms/step - loss: 0.2772 - accuracy:
0.9057 - val_loss: 1.8266 - val_accuracy: 0.5409
Epoch 50/50
1390/1390 [==============================] - 32s 23ms/step - loss: 0.2768 - accuracy:
0.9062 - val_loss: 1.8164 - val_accuracy: 0.5399
```

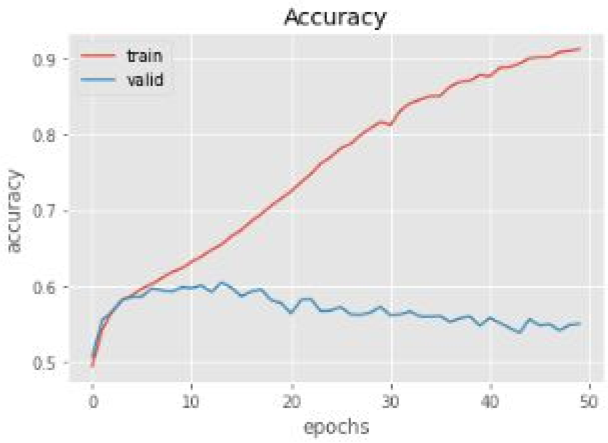




```
from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(layers.LSTM(128))
model.add(Dropout(0.25))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
   x= x_train, y=y_train, batch_size=32, epochs=50, validation_data=(x_val, y_val)
)
```

```
1390/1390 [==============================] - 29s 21ms/step - loss: 0.2932 - accuracy: 0.9019 - val_loss:
1.8266 - val_accuracy: 0.5493
Epoch 48/50
1390/1390 [==============================] - 29s 21ms/step - loss: 0.2722 - accuracy: 0.9085 - val_loss:
1.9326 - val_accuracy: 0.5414
Epoch 49/50
1390/1390 [==============================] - 29s 21ms/step - loss: 0.2697 - accuracy: 0.9101 - val_loss:
1.9324 - val_accuracy: 0.5481
Epoch 50/50
1390/1390 [==============================] - 29s 21ms/step - loss: 0.2636 - accuracy: 0.9122 - val_loss:
1.9015 - val_accuracy: 0.5499
```





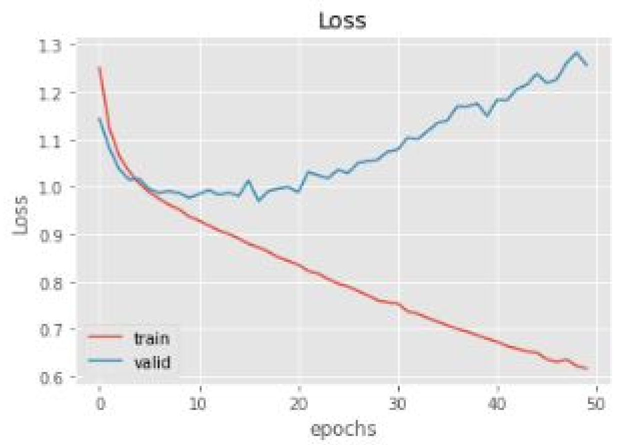
```
from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(layers.LSTM(128))
model.add(Dropout(0.50))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
   x= x_train, y=y_train, batch_size=32, epochs=50, validation_data=(x_val, y_val)
```

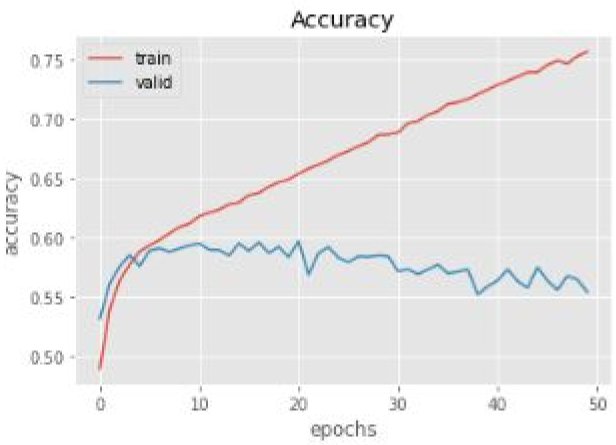
```
Epoch 1/50
1390/1390 [==============================] - 30s 22ms/step - loss: 1.2533 - accuracy:
0.4989 - val_loss: 1.1596 - val_accuracy: 0.5241
Epoch 2/50
1390/1390 [==============================] - 28s 20ms/step - loss: 1.1338 - accuracy:
0.5364 - val_loss: 1.1028 - val_accuracy: 0.5547
Epoch 3/50
1390/1390 [==============================] - 28s 20ms/step - loss: 1.0748 - accuracy:
0.5625 - val_loss: 1.0230 - val_accuracy: 0.5820
Epoch 4/50
1390/1390 [==============================] - 28s 20ms/step - loss: 1.0351 - accuracy:
0.5757 - val_loss: 1.0124 - val_accuracy: 0.5825
Epoch 5/50
1390/1390 [==============================] - 28s 20ms/step - loss: 1.0088 - accuracy:
0.5879 - val_loss: 1.0011 - val_accuracy: 0.5918
Epoch 6/50
1390/1390 [==============================] - 29s 21ms/step - loss: 0.9920 - accuracy:
0.5965 - val_loss: 0.9917 - val_accuracy: 0.5851
Epoch 7/50
1390/1390 [==============================] - 29s 21ms/step - loss: 0.9712 - accuracy:
0.6027 - val_loss: 1.0001 - val_accuracy: 0.5892
Epoch 8/50
1390/1390 [==============================] - 30s 22ms/step - loss: 0.9577 - accuracy:
0.6090 - val_loss: 0.9850 - val_accuracy: 0.5868
Epoch 9/50
1390/1390 [==============================] - 30s 21ms/step - loss: 0.9418 - accuracy:
0.6134 - val_loss: 0.9773 - val_accuracy: 0.5958
```

```
Epoch 10/50
1390/1390 [==============================] - 31s 22ms/step - loss: 0.9243 - accuracy:
0.6191 - val_loss: 0.9861 - val_accuracy: 0.6004
Epoch 11/50
1390/1390 [==============================] - 28s 20ms/step - loss: 0.9094 - accuracy:
0.6260 - val_loss: 0.9799 - val_accuracy: 0.5846
Epoch 12/50
1390/1390 [==============================] - 29s 21ms/step - loss: 0.8937 - accuracy:
0.6310 - val_loss: 0.9764 - val_accuracy: 0.5978
Epoch 13/50
1390/1390 [==============================] - 28s 20ms/step - loss: 0.8767 - accuracy:
0.6369 - val_loss: 0.9741 - val_accuracy: 0.5984
Epoch 14/50
1390/1390 [==============================] - 29s 21ms/step - loss: 0.8576 - accuracy:
0.6452 - val_loss: 0.9985 - val_accuracy: 0.5835
Epoch 15/50
1390/1390 [==============================] - 30s 21ms/step - loss: 0.8377 - accuracy:
0.6527 - val_loss: 1.0061 - val_accuracy: 0.5901
Epoch 16/50
1390/1390 [==============================] - 29s 21ms/step - loss: 0.8189 - accuracy:
0.6626 - val_loss: 1.0037 - val_accuracy: 0.5956
Epoch 17/50
1390/1390 [==============================] - 29s 21ms/step - loss: 0.7993 - accuracy:
0.6693 - val_loss: 1.0206 - val_accuracy: 0.5904
Epoch 18/50
 159/1390 [==>...........................] - ETA: 26s - loss: 0.7758 - accuracy: 0.6781
```

```
from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(layers.LSTM(64))
#model.add(Dropout(0.50))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
    x= x_train, y=y_train, batch_size=32, epochs=50, validation_data=(x_val, y_val)
)
```

```
Epoch 46/50
1390/1390 [==============================] - 18s 13ms/step - loss: 0.6373 - accuracy:
0.7455 - val_loss: 1.2197 - val_accuracy: 0.5640
Epoch 47/50
1390/1390 [==============================] - 18s 13ms/step - loss: 0.6311 - accuracy:
0.7489 - val_loss: 1.2272 - val_accuracy: 0.5563
Epoch 48/50
1390/1390 [==============================] - 18s 13ms/step - loss: 0.6363 - accuracy:
0.7464 - val_loss: 1.2621 - val_accuracy: 0.5678
Epoch 49/50
1390/1390 [==============================] - 18s 13ms/step - loss: 0.6231 - accuracy:
0.7527 - val_loss: 1.2838 - val_accuracy: 0.5649
Epoch 50/50
1390/1390 [==============================] - 18s 13ms/step - loss: 0.6176 - accuracy:
0.7564 - val_loss: 1.2576 - val_accuracy: 0.5545
```





```
from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(layers.LSTM(128))
model.add(Dropout(0.50))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
    x= x_train, y=y_train, batch_size=64, epochs=50, validation_data=(x_val, y_val)

695/695 [==============================] - 18s 25ms/step - loss: 0.9162 - accuracy:
0.6222 - val_loss: 0.9830 - val_accuracy: 0.5971
Epoch 14/50
695/695 [==============================] - 18s 25ms/step - loss: 0.9042 - accuracy:
0.6274 - val_loss: 0.9781 - val_accuracy: 0.5966
Epoch 15/50
695/695 [==============================] - 18s 26ms/step - loss: 0.8916 - accuracy:
0.6305 - val_loss: 1.0050 - val_accuracy: 0.5934
Epoch 16/50
695/695 [==============================] - 18s 26ms/step - loss: 0.8786 - accuracy:
0.6363 - val_loss: 0.9949 - val_accuracy: 0.5978
```

```
from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(layers.LSTM(128))
model.add(Dropout(0.50))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
    x= x_train, y=y_train, batch_size=16, epochs=50, validation_data=(x_val, y_val))
Epoch 1/50
2779/2779 [==============================] - 51s 18ms/step - loss: 1.2849 - accuracy:
0.4883 - val_loss: 1.1549 - val_accuracy: 0.5308
Epoch 2/50
2779/2779 [==============================] - 51s 18ms/step - loss: 1.1381 - accuracy:
0.5371 - val_loss: 1.0633 - val_accuracy: 0.5655
Epoch 3/50
2779/2779 [==============================] - 51s 18ms/step - loss: 1.0699 - accuracy:
0.5663 - val_loss: 1.0289 - val_accuracy: 0.5702
Epoch 4/50
2779/2779 [==============================] - 51s 18ms/step - loss: 1.0288 - accuracy:
0.5814 - val_loss: 1.0229 - val_accuracy: 0.5797
Epoch 5/50
2779/2779 [==============================] - 51s 18ms/step - loss: 1.0026 - accuracy:
0.5918 - val_loss: 0.9887 - val_accuracy: 0.5890
Epoch 6/50
2779/2779 [==============================] - 51s 18ms/step - loss: 0.9754 - accuracy:
0.5997 - val_loss: 0.9803 - val_accuracy: 0.5924
Epoch 7/50
2779/2779 [==============================] - 51s 18ms/step - loss: 0.9542 - accuracy:
0.6059 - val_loss: 0.9948 - val_accuracy: 0.5842
Epoch 8/50
2779/2779 [==============================] - 51s 18ms/step - loss: 0.9341 - accuracy:
0.6146 - val_loss: 0.9818 - val_accuracy: 0.5958
Epoch 9/50
2779/2779 [==============================] - 51s 18ms/step - loss: 0.9123 - accuracy:
0.6244 - val_loss: 0.9874 - val_accuracy: 0.5955
Epoch 10/50
2779/2779 [==============================] - 51s 18ms/step - loss: 0.8919 - accuracy:
0.6320 - val_loss: 0.9748 - val_accuracy: 0.5982
Epoch 11/50
2779/2779 [==============================] - 51s 18ms/step - loss: 0.8692 - accuracy:
0.6411 - val_loss: 0.9874 - val_accuracy: 0.5889
Epoch 12/50
2779/2779 [==============================] - 51s 18ms/step - loss: 0.8440 - accuracy:
0.6531 - val_loss: 0.9731 - val_accuracy: 0.5993
Epoch 13/50
2779/2779 [==============================] - 51s 18ms/step - loss: 0.8192 - accuracy:
0.6636 - val_loss: 0.9956 - val_accuracy: 0.5914
Epoch 14/50
2779/2779 [==============================] - 51s 18ms/step - loss: 0.7921 - accuracy:
0.6770 - val_loss: 1.0043 - val_accuracy: 0.5944
Epoch 15/50
```

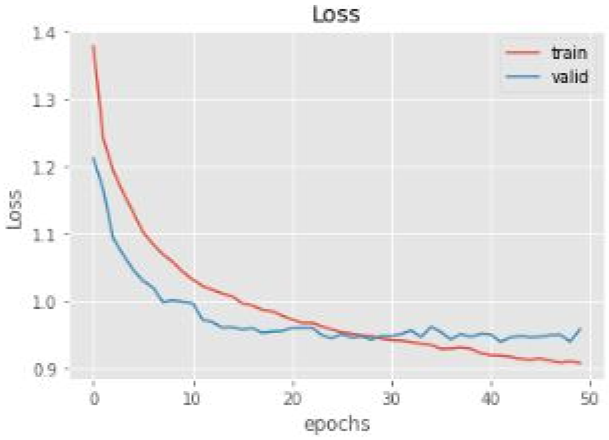
```
2779/2779 [==============================] - 53s 19ms/step - loss: 0.7645 - accuracy:
0.6870 - val_loss: 1.0337 - val_accuracy: 0.5908
Epoch 16/50
2779/2779 [==============================] - 52s 19ms/step - loss: 0.7340 - accuracy:
0.6989 - val_loss: 1.0781 - val_accuracy: 0.5844
Epoch 17/50
2779/2779 [==============================] - 51s 18ms/step - loss: 0.7073 - accuracy:
0.7127 - val_loss: 1.1340 - val_accuracy: 0.5806
Epoch 18/50
2779/2779 [==============================] - 51s 18ms/step - loss: 0.6733 - accuracy:
0.7281 - val_loss: 1.1734 - val_accuracy: 0.5836
Epoch 19/50
2779/2779 [==============================] - 51s 18ms/step - loss: 0.6501 - accuracy:
0.7405 - val_loss: 1.1881 - val_accuracy: 0.5640
Epoch 20/50
2779/2779 [==============================] - 51s 18ms/step - loss: 0.6215 - accuracy:
0.7538 - val_loss: 1.2116 - val_accuracy: 0.5770
Epoch 21/50
2779/2779 [==============================] - 51s 18ms/step - loss: 0.5946 - accuracy:
0.7689 - val_loss: 1.2390 - val_accuracy: 0.5591
Epoch 22/50
2779/2779 [==============================] - 51s 18ms/step - loss: 0.5699 - accuracy:
0.7780 - val_loss: 1.2758 - val_accuracy: 0.5588
Epoch 23/50
2779/2779 [==============================] - 51s 18ms/step - loss: 0.5396 - accuracy:
0.7927 - val_loss: 1.3430 - val_accuracy: 0.5749
Epoch 24/50
2779/2779 [==============================] - 52s 19ms/step - loss: 0.5158 - accuracy:
0.8056 - val_loss: 1.3545 - val_accuracy: 0.5475
Epoch 25/50
2779/2779 [==============================] - 50s 18ms/step - loss: 0.4875 - accuracy:
0.8175 - val_loss: 1.4000 - val_accuracy: 0.5461
Epoch 26/50
2779/2779 [==============================] - 48s 17ms/step - loss: 0.4667 - accuracy:
0.8265 - val_loss: 1.4322 - val_accuracy: 0.5614
Epoch 27/50
2779/2779 [==============================] - 48s 17ms/step - loss: 0.4501 - accuracy:
0.8368 - val_loss: 1.4231 - val_accuracy: 0.5458
Epoch 28/50
2779/2779 [==============================] - 48s 17ms/step - loss: 0.4292 - accuracy:
0.8446 - val_loss: 1.5180 - val_accuracy: 0.5510
Epoch 29/50
2779/2779 [==============================] - 49s 18ms/step - loss: 0.4091 - accuracy:
0.8540 - val_loss: 1.5464 - val_accuracy: 0.5436
```

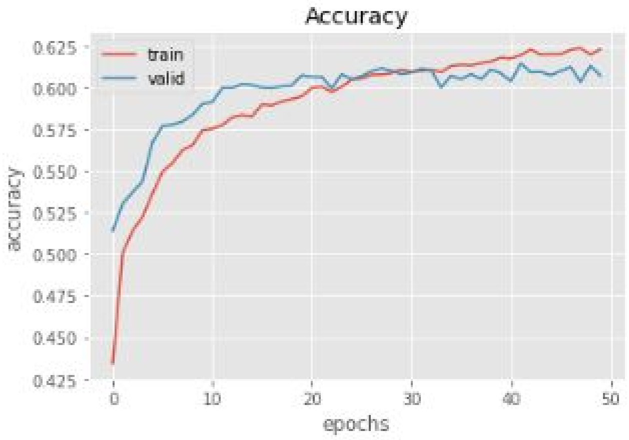
```
Epoch 30/50
2779/2779 [==============================] - 48s 17ms/step - loss: 0.3959 - accuracy:
0.8591 - val_loss: 1.6358 - val_accuracy: 0.5531
Epoch 31/50
2779/2779 [==============================] - 48s 17ms/step - loss: 0.3865 - accuracy:
0.8636 - val_loss: 1.6106 - val_accuracy: 0.5345
Epoch 32/50
2779/2779 [==============================] - 48s 17ms/step - loss: 0.3624 - accuracy:
0.8743 - val_loss: 1.7052 - val_accuracy: 0.5531
Epoch 33/50
2779/2779 [==============================] - 48s 17ms/step - loss: 0.3626 - accuracy:
0.8741 - val_loss: 1.6298 - val_accuracy: 0.5401
Epoch 34/50
2779/2779 [==============================] - 48s 17ms/step - loss: 0.3497 - accuracy:
0.8787 - val_loss: 1.6910 - val_accuracy: 0.5389
Epoch 35/50
2779/2779 [==============================] - 52s 19ms/step - loss: 0.3290 - accuracy:
0.8871 - val_loss: 1.7963 - val_accuracy: 0.5447
Epoch 36/50
2779/2779 [==============================] - 50s 18ms/step - loss: 0.3265 - accuracy:
0.8891 - val_loss: 1.7917 - val_accuracy: 0.5268
Epoch 37/50
 209/2779 [=>............................] - ETA: 46s - loss: 0.2895 - accuracy: 0.9097
```

```
from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(keras.layers.LSTM(128, dropout=0.5))
#model.add(Dropout(0.50))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
    x= x_train, y=y_train, batch_size=16, epochs=50, validation_data=(x_val, y_val))
```

```
779/2779 [==============================] - 49s 18ms/step - loss: 0.9140 - accuracy:
0.6199 - val_loss: 0.9462 - val_accuracy: 0.6098
```

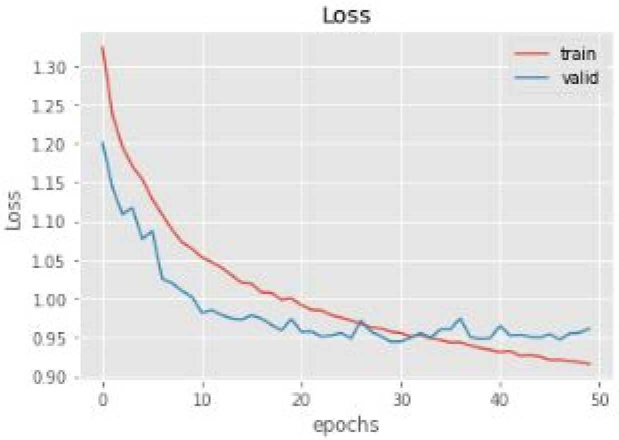
```
Epoch 47/50
2779/2779 [==============================] - 49s 18ms/step - loss: 0.9106 - accuracy:
0.6226 - val_loss: 0.9488 - val_accuracy: 0.6123
Epoch 48/50
2779/2779 [==============================] - 49s 18ms/step - loss: 0.9074 - accuracy:
0.6236 - val_loss: 0.9493 - val_accuracy: 0.6034
Epoch 49/50
2779/2779 [==============================] - 49s 18ms/step - loss: 0.9096 - accuracy:
0.6197 - val_loss: 0.9391 - val_accuracy: 0.6130
Epoch 50/50
2779/2779 [==============================] - 49s 18ms/step - loss: 0.9068 - accuracy:
0.6229 - val_loss: 0.9575 - val_accuracy: 0.6070
```

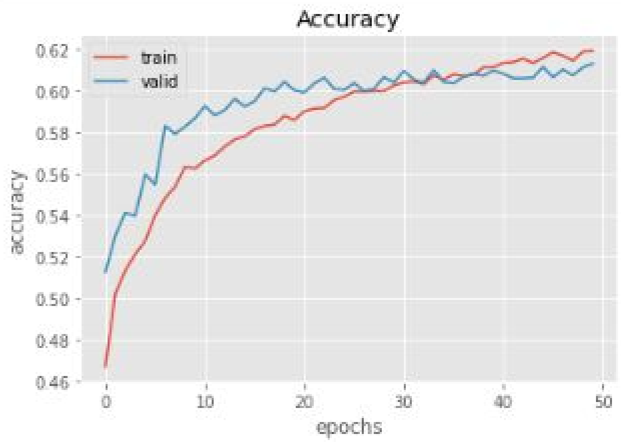




```
from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(keras.layers.LSTM(128, dropout=0.5))
#model.add(Dropout(0.50))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
   x= x_train, y=y_train, batch_size=32, epochs=50, validation_data=(x_val, y_val))
```

```
1390/1390 [==============================] - 42s 30ms/step - loss: 0.9211 - accuracy:
0.6188 - val_loss: 0.9545 - val_accuracy: 0.6066
Epoch 47/50
1390/1390 [==============================] - 41s 30ms/step - loss: 0.9213 - accuracy:
0.6170 - val_loss: 0.9474 - val_accuracy: 0.6104
Epoch 48/50
1390/1390 [==============================] - 42s 30ms/step - loss: 0.9198 - accuracy:
0.6146 - val_loss: 0.9553 - val_accuracy: 0.6076
Epoch 49/50
1390/1390 [==============================] - 43s 31ms/step - loss: 0.9183 - accuracy:
0.6190 - val_loss: 0.9561 - val_accuracy: 0.6114
Epoch 50/50
1390/1390 [==============================] - 42s 30ms/step - loss: 0.9158 - accuracy:
0.6193 - val_loss: 0.9615 - val_accuracy: 0.6133
```





```
from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)

RANDOM_SEED = 42
x_train =  [ emb for emb in b2wCorpusTrain.review_text]
y_train =   b2wCorpusTrain.overall_rating
x_val = [ emb for emb in b2wCorpusValidate.review_text ]
y_val = b2wCorpusValidate.overall_rating
x_train = np.asarray(x_train)
x_val =np.asarray(x_val)

np.random.seed(RANDOM_SEED)
model = KeyedVectors.load_word2vec_format('word2vec_200k.txt')
emb = model.get_keras_embedding()
emb.trainable = True

model.add(keras.layers.LSTM(128, dropout=0.5))
#model.add(Dropout(0.50))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
   x= x_train, y=y_train, batch_size=32, epochs=50, validation_data=(x_val, y_val))
```

```
Epoch 1/50
1390/1390 [==============================] - 200s 144ms/step - loss: 1.1576 -
accuracy: 0.5277 - val_loss: 1.0475 - val_accuracy: 0.5661
Epoch 2/50
1390/1390 [==============================] - 169s 122ms/step - loss: 1.0330 -
accuracy: 0.5646 - val_loss: 0.9862 - val_accuracy: 0.5933
Epoch 3/50
1390/1390 [==============================] - 180s 130ms/step - loss: 0.9777 -
accuracy: 0.5922 - val_loss: 0.9954 - val_accuracy: 0.5968
Epoch 4/50
1390/1390 [==============================] - 179s 128ms/step - loss: 0.9275 -
accuracy: 0.6126 - val_loss: 0.9401 - val_accuracy: 0.6186
Epoch 5/50
1390/1390 [==============================] - 179s 129ms/step - loss: 0.8927 -
accuracy: 0.6262 - val_loss: 0.9241 - val_accuracy: 0.6199
Epoch 6/50
1390/1390 [==============================] - 181s 130ms/step - loss: 0.8662 -
accuracy: 0.6405 - val_loss: 0.9374 - val_accuracy: 0.6102
Epoch 7/50
1390/1390 [==============================] - 180s 129ms/step - loss: 0.8375 -
accuracy: 0.6508 - val_loss: 0.9213 - val_accuracy: 0.6167
Epoch 8/50
1390/1390 [==============================] - 195s 140ms/step - loss: 0.8099 -
accuracy: 0.6637 - val_loss: 0.9341 - val_accuracy: 0.6202
Epoch 9/50
1390/1390 [==============================] - 203s 146ms/step - loss: 0.7839 -
accuracy: 0.6753 - val_loss: 0.9330 - val_accuracy: 0.6187
Epoch 10/50
 770/1390 [===============>..............] - ETA: 1:29 - loss: 0.7569 - accuracy: 0.6897
```

```
from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)

model.add(keras.layers.LSTM(128, dropout=0.5))
#model.add(Dropout(0.50))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
```

```
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
    x= x_train, y=y_train, batch_size=64, epochs=50, validation_data=(x_val, y_val))
Epoch 1/50
695/695 [==============================] - 101s 145ms/step - loss: 0.9528 - accuracy:
0.6048 - val_loss: 0.9637 - val_accuracy: 0.6101
Epoch 2/50
695/695 [==============================] - 73s 105ms/step - loss: 0.8155 - accuracy:
0.6637 - val_loss: 0.9636 - val_accuracy: 0.6038
Epoch 3/50
695/695 [==============================] - 72s 103ms/step - loss: 0.7811 - accuracy:
0.6805 - val_loss: 0.9802 - val_accuracy: 0.6137
Epoch 4/50
695/695 [==============================] - 72s 104ms/step - loss: 0.7551 - accuracy:
0.6918 - val_loss: 0.9991 - val_accuracy: 0.6083
Epoch 5/50
 66/695 [=>............................] - ETA: 1:05 - loss: 0.7161 - accuracy: 0.7114

from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)

model.add(keras.layers.LSTM(64, dropout=0.5))
#model.add(Dropout(0.50))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
    x= x_train, y=y_train, batch_size=64, epochs=50, validation_data=(x_val, y_val))
Epoch 1/50
695/695 [==============================] - 68s 98ms/step - loss: 0.9385 - accuracy:
0.6124 - val_loss: 0.9941 - val_accuracy: 0.5978
Epoch 2/50
695/695 [==============================] - 68s 98ms/step - loss: 0.7855 - accuracy:
0.6819 - val_loss: 1.0176 - val_accuracy: 0.6089
Epoch 3/50
```

```
 89/695 [==>...........................] - ETA: 59s - loss: 0.7612 - accuracy: 0.6914

from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)

model.add(keras.layers.LSTM(32, dropout=0.5))
#model.add(Dropout(0.50))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
    x= x_train, y=y_train, batch_size=64, epochs=50, validation_data=(x_val, y_val))
Epoch 1/50
695/695 [==============================] - 68s 97ms/step - loss: 0.9897 - accuracy:
0.5850 - val_loss: 1.0126 - val_accuracy: 0.5972
Epoch 2/50
695/695 [==============================] - 70s 100ms/step - loss: 0.8125 - accuracy:
0.6753 - val_loss: 1.0194 - val_accuracy: 0.5901
Epoch 3/50
695/695 [==============================] - 72s 103ms/step - loss: 0.7854 - accuracy:
0.6902 - val_loss: 1.0202 - val_accuracy: 0.6001
Epoch 4/50
583/695 [========================>.....] - ETA: 11s - loss: 0.7559 - accuracy: 0.7008

from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)

model.add(keras.layers.LSTM(16, dropout=0.5))
#model.add(Dropout(0.50))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
```

```
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
    x= x_train, y=y_train, batch_size=64, epochs=50, validation_data=(x_val, y_val))
Epoch 1/50
695/695 [==============================] - 72s 104ms/step - loss: 1.1259 - accuracy:
0.5459 - val_loss: 1.0366 - val_accuracy: 0.5716
Epoch 2/50
695/695 [==============================] - 72s 104ms/step - loss: 0.8588 - accuracy:
0.6459 - val_loss: 1.0240 - val_accuracy: 0.5841
Epoch 3/50
695/695 [==============================] - 66s 96ms/step - loss: 0.7809 - accuracy:
0.6866 - val_loss: 1.0423 - val_accuracy: 0.5860
Epoch 4/50
383/695 [================>.............] - ETA: 29s - loss: 0.7448 - accuracy: 0.7016
```

```
model.add(keras.layers.LSTM(16, dropout=0.5))
#model.add(Dropout(0.50))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
    x= x_train, y=y_train, batch_size=32, epochs=50, validation_data=(x_val, y_val))
Epoch 1/50
1390/1390 [==============================] - 131s 94ms/step - loss: 1.2666 - accuracy:
0.4678 - val_loss: 1.0826 - val_accuracy: 0.5418
Epoch 2/50
1390/1390 [==============================] - 132s 95ms/step - loss: 1.0312 - accuracy:
0.5724 - val_loss: 1.0111 - val_accuracy: 0.5882
Epoch 3/50
1390/1390 [==============================] - 132s 95ms/step - loss: 0.9715 - accuracy:
0.5985 - val_loss: 0.9543 - val_accuracy: 0.6101
Epoch 4/50
1390/1390 [==============================] - 135s 97ms/step - loss: 0.9361 - accuracy:
0.6119 - val_loss: 0.9471 - val_accuracy: 0.6132
Epoch 5/50
1390/1390 [==============================] - 133s 96ms/step - loss: 0.9036 - accuracy:
0.6218 - val_loss: 0.9524 - val_accuracy: 0.6126
Epoch 6/50
```

```
1390/1390 [==============================] - 133s 96ms/step - loss: 0.8772 - accuracy:
0.6334 - val_loss: 0.9495 - val_accuracy: 0.6129
Epoch 7/50
1390/1390 [==============================] - 139s 100ms/step - loss: 0.8528 -
accuracy: 0.6457 - val_loss: 0.9534 - val_accuracy: 0.6145
Epoch 8/50
 263/1390 [====>.........................] - ETA: 1:48 - loss: 0.8176 - accuracy: 0.6568
```

```
from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(keras.layers.LSTM(16, dropout=0.5, return_sequences=True))
#model.add(keras.layers.LSTM(16, kernel_regularizer=l2(0.01), recurrent_regularizer=l2(0.01),
bias_regularizer=l2(0.01)))
model.add(keras.layers.LSTM(16, dropout=0.5))
#model.add(Dropout(0.50))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
   x= x_train, y=y_train, batch_size=32, epochs=50, validation_data=(x_val, y_val))
Epoch 1/50
1390/1390 [==============================] - 133s 96ms/step - loss: 1.0310 - accuracy:
0.5801 - val_loss: 1.0027 - val_accuracy: 0.5999
Epoch 2/50
1390/1390 [==============================] - 136s 98ms/step - loss: 0.8765 - accuracy:
0.6437 - val_loss: 0.9978 - val_accuracy: 0.6056
Epoch 3/50
1390/1390 [==============================] - 134s 97ms/step - loss: 0.8397 - accuracy:
0.6569 - val_loss: 0.9787 - val_accuracy: 0.6073
Epoch 4/50
1390/1390 [==============================] - 140s 101ms/step - loss: 0.8172 -
accuracy: 0.6673 - val_loss: 1.0083 - val_accuracy: 0.5946
Epoch 5/50
1390/1390 [==============================] - 147s 106ms/step - loss: 0.7929 -
accuracy: 0.6783 - val_loss: 1.0111 - val_accuracy: 0.6028
Epoch 6/50
```

```
1390/1390 [==============================] - 139s 100ms/step - loss: 0.7769 -
accuracy: 0.6861 - val_loss: 1.0185 - val_accuracy: 0.6022
Epoch 7/50
1390/1390 [==============================] - 135s 97ms/step - loss: 0.7592 - accuracy:
0.6915 - val_loss: 1.0185 - val_accuracy: 0.5961
Epoch 8/50
1017/1390 [====================>.........] - ETA: 35s - loss: 0.7378 - accuracy: 0.7024
```

```
rom tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(keras.layers.LSTM(16, dropout=0.5, return_sequences=True))
model.add(keras.layers.LSTM(16, kernel_regularizer=l2(0.01), recurrent_regularizer=l2(0.01),
bias_regularizer=l2(0.01)))
#model.add(keras.layers.LSTM(16, dropout=0.5))
#model.add(Dropout(0.50))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
   x= x_train, y=y_train, batch_size=32, epochs=50, validation_data=(x_val, y_val))
from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(keras.layers.LSTM(16, dropout=0.5, return_sequences=True))
model.add(keras.layers.LSTM(16, kernel_regularizer=l2(0.01), recurrent_regularizer=l2(0.01),
bias_regularizer=l2(0.01)))
#model.add(keras.layers.LSTM(16, dropout=0.5))
#model.add(Dropout(0.50))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
```

```
history = model.fit(
    x= x_train, y=y_train, batch_size=32, epochs=50, validation_data=(x_val, y_val))
Epoch 1/50
1390/1390 [==============================] - 133s 96ms/step - loss: 1.2651 - accuracy:
0.5452 - val_loss: 1.1904 - val_accuracy: 0.5525
Epoch 2/50
1390/1390 [==============================] - 133s 96ms/step - loss: 1.0919 - accuracy:
0.5691 - val_loss: 1.1078 - val_accuracy: 0.5694
Epoch 3/50
1390/1390 [==============================] - 132s 95ms/step - loss: 0.9393 - accuracy:
0.6207 - val_loss: 1.0436 - val_accuracy: 0.5977
Epoch 4/50
1390/1390 [==============================] - 125s 90ms/step - loss: 0.8446 - accuracy:
0.6741 - val_loss: 1.0533 - val_accuracy: 0.5956
Epoch 5/50
1390/1390 [==============================] - 125s 90ms/step - loss: 0.7935 - accuracy:
0.6944 - val_loss: 1.0595 - val_accuracy: 0.5867
Epoch 6/50
 630/1390 [============>.................] - ETA: 1:09 - loss: 0.7571 - accuracy: 0.7128

from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(keras.layers.LSTM(16, dropout=0.25, return_sequences=True))
#model.add(keras.layers.LSTM(16, kernel_regularizer=l2(0.01), recurrent_regularizer=l2(0.01),
bias_regularizer=l2(0.01)))
model.add(keras.layers.LSTM(16, dropout=0.25))
#model.add(Dropout(0.50))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
    x= x_train, y=y_train, batch_size=16, epochs=50, validation_data=(x_val, y_val))
Epoch 1/50
2779/2779 [==============================] - 257s 93ms/step - loss: 0.8913 - accuracy:
0.6409 - val_loss: 1.0764 - val_accuracy: 0.5841
Epoch 2/50
```

```
2779/2779 [==============================] - 252s 91ms/step - loss: 0.7406 - accuracy:
0.7101 - val_loss: 1.0885 - val_accuracy: 0.5823
Epoch 3/50
2779/2779 [==============================] - 252s 91ms/step - loss: 0.6958 - accuracy:
0.7299 - val_loss: 1.1163 - val_accuracy: 0.5871
Epoch 4/50
2779/2779 [==============================] - 252s 91ms/step - loss: 0.6633 - accuracy:
0.7452 - val_loss: 1.1257 - val_accuracy: 0.5781
Epoch 5/50
2779/2779 [==============================] - 252s 91ms/step - loss: 0.6282 - accuracy:
0.7618 - val_loss: 1.1829 - val_accuracy: 0.5762
Epoch 6/50
2779/2779 [==============================] - 252s 91ms/step - loss: 0.5998 - accuracy:
0.7737 - val_loss: 1.2272 - val_accuracy: 0.5800
Epoch 7/50
2779/2779 [==============================] - 252s 91ms/step - loss: 0.5725 - accuracy:
0.7847 - val_loss: 1.2578 - val_accuracy: 0.5734
Epoch 8/50
2779/2779 [==============================] - 253s 91ms/step - loss: 0.5525 - accuracy:
0.7924 - val_loss: 1.2674 - val_accuracy: 0.5661
Epoch 9/50
2779/2779 [==============================] - 253s 91ms/step - loss: 0.5261 - accuracy:
0.8025 - val_loss: 1.3053 - val_accuracy: 0.5725
Epoch 10/50
2779/2779 [==============================] - 252s 91ms/step - loss: 0.5105 - accuracy:
0.8072 - val_loss: 1.3188 - val_accuracy: 0.5713
Epoch 11/50
2779/2779 [==============================] - 253s 91ms/step - loss: 0.4882 - accuracy:
0.8170 - val_loss: 1.3507 - val_accuracy: 0.5791
Epoch 12/50
2779/2779 [==============================] - 254s 91ms/step - loss: 0.4705 - accuracy:
0.8221 - val_loss: 1.4410 - val_accuracy: 0.5699
Epoch 13/50
2779/2779 [==============================] - 253s 91ms/step - loss: 0.4572 - accuracy:
0.8254 - val_loss: 1.4931 - val_accuracy: 0.5693
Epoch 14/50
2779/2779 [==============================] - 253s 91ms/step - loss: 0.4388 - accuracy:
0.8323 - val_loss: 1.4727 - val_accuracy: 0.5759
Epoch 15/50
2779/2779 [==============================] - 253s 91ms/step - loss: 0.4250 - accuracy:
0.8354 - val_loss: 1.4816 - val_accuracy: 0.5659
Epoch 16/50
2779/2779 [==============================] - 253s 91ms/step - loss: 0.4144 - accuracy:
0.8410 - val_loss: 1.5340 - val_accuracy: 0.5722
```

```
Epoch 17/50
2779/2779 [==============================] - 253s 91ms/step - loss: 0.4010 - accuracy:
0.8468 - val_loss: 1.5853 - val_accuracy: 0.5711
Epoch 18/50
2779/2779 [==============================] - 253s 91ms/step - loss: 0.3899 - accuracy:
0.8505 - val_loss: 1.6548 - val_accuracy: 0.5690
Epoch 19/50
2779/2779 [==============================] - 253s 91ms/step - loss: 0.3793 - accuracy:
0.8527 - val_loss: 1.5753 - val_accuracy: 0.5693
Epoch 20/50
2779/2779 [==============================] - 253s 91ms/step - loss: 0.3666 - accuracy:
0.8585 - val_loss: 1.7306 - val_accuracy: 0.5702
Epoch 21/50
2779/2779 [==============================] - 253s 91ms/step - loss: 0.3599 - accuracy:
0.8598 - val_loss: 1.6544 - val_accuracy: 0.5662
Epoch 22/50
2779/2779 [==============================] - 253s 91ms/step - loss: 0.3470 - accuracy:
0.8660 - val_loss: 1.7737 - val_accuracy: 0.5605
Epoch 23/50
2779/2779 [==============================] - 253s 91ms/step - loss: 0.3421 - accuracy:
0.8676 - val_loss: 1.7171 - val_accuracy: 0.5667
Epoch 24/50
2779/2779 [==============================] - 254s 91ms/step - loss: 0.3317 - accuracy:
0.8709 - val_loss: 1.8208 - val_accuracy: 0.5640
Epoch 25/50
2779/2779 [==============================] - 254s 91ms/step - loss: 0.3254 - accuracy:
0.8729 - val_loss: 1.8016 - val_accuracy: 0.5702
Epoch 26/50
2779/2779 [==============================] - 254s 91ms/step - loss: 0.3168 - accuracy:
0.8763 - val_loss: 1.8495 - val_accuracy: 0.5607
Epoch 27/50
2779/2779 [==============================] - 254s 91ms/step - loss: 0.3106 - accuracy:
0.8783 - val_loss: 1.8282 - val_accuracy: 0.5659
Epoch 28/50
2779/2779 [==============================] - 253s 91ms/step - loss: 0.3001 - accuracy:
0.8831 - val_loss: 1.8952 - val_accuracy: 0.5605
Epoch 29/50
2779/2779 [==============================] - 256s 92ms/step - loss: 0.2980 - accuracy:
0.8851 - val_loss: 1.8891 - val_accuracy: 0.5620
Epoch 30/50
2779/2779 [==============================] - 258s 93ms/step - loss: 0.2883 - accuracy:
0.8899 - val_loss: 1.8819 - val_accuracy: 0.5516
Epoch 31/50
```

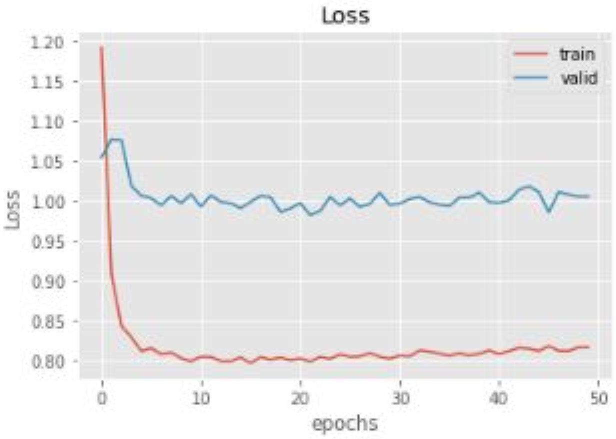
```
2779/2779 [==============================] - 253s 91ms/step - loss: 0.2850 - accuracy:
0.8912 - val_loss: 1.8918 - val_accuracy: 0.5610
Epoch 32/50
2779/2779 [==============================] - 268s 96ms/step - loss: 0.2808 - accuracy:
0.8923 - val_loss: 1.9029 - val_accuracy: 0.5596
Epoch 33/50
2779/2779 [==============================] - 272s 98ms/step - loss: 0.2697 - accuracy:
0.8959 - val_loss: 2.0362 - val_accuracy: 0.5553
Epoch 34/50
2779/2779 [==============================] - 268s 97ms/step - loss: 0.2714 - accuracy:
0.8963 - val_loss: 1.9389 - val_accuracy: 0.5617
Epoch 35/50
2779/2779 [==============================] - 269s 97ms/step - loss: 0.2656 - accuracy:
0.8979 - val_loss: 1.9981 - val_accuracy: 0.5522
Epoch 36/50
2779/2779 [==============================] - 269s 97ms/step - loss: 0.2570 - accuracy:
0.9010 - val_loss: 2.0917 - val_accuracy: 0.5516
Epoch 37/50
2779/2779 [==============================] - 269s 97ms/step - loss: 0.2554 - accuracy:
0.9015 - val_loss: 2.0042 - val_accuracy: 0.5522
```

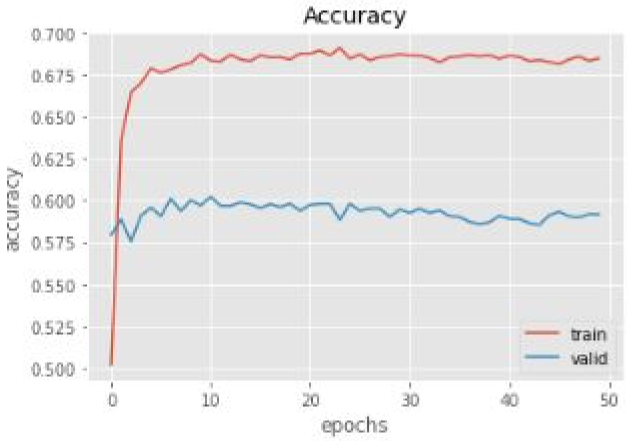
```
from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(keras.layers.LSTM(16, dropout=0.8, return_sequences=True))
#model.add(keras.layers.LSTM(16, kernel_regularizer=l2(0.01), recurrent_regularizer=l2(0.01),
bias_regularizer=l2(0.01)))
model.add(keras.layers.LSTM(16, dropout=0.8))
#model.add(Dropout(0.50))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
   x= x_train, y=y_train, batch_size=64, epochs=50, validation_data=(x_val, y_val))
Epoch 1/50
695/695 [==============================] - 72s 103ms/step - loss: 1.3385 - accuracy:
0.4519 - val_loss: 1.1537 - val_accuracy: 0.5516
Epoch 2/50
```

```
695/695 [==============================] - 75s 107ms/step - loss: 1.0801 - accuracy:
0.5558 - val_loss: 1.1322 - val_accuracy: 0.5614
Epoch 3/50
695/695 [==============================] - 80s 115ms/step - loss: 0.9824 - accuracy:
0.5897 - val_loss: 1.0992 - val_accuracy: 0.5832
Epoch 4/50
695/695 [==============================] - 81s 116ms/step - loss: 0.9117 - accuracy:
0.6295 - val_loss: 1.0716 - val_accuracy: 0.5931
Epoch 5/50
695/695 [==============================] - 80s 115ms/step - loss: 0.8717 - accuracy:
0.6527 - val_loss: 1.0440 - val_accuracy: 0.5924
Epoch 6/50
695/695 [==============================] - 80s 115ms/step - loss: 0.8383 - accuracy:
0.6638 - val_loss: 1.0655 - val_accuracy: 0.5952
Epoch 7/50
695/695 [==============================] - 78s 113ms/step - loss: 0.8082 - accuracy:
0.6777 - val_loss: 1.0682 - val_accuracy: 0.5959
Epoch 8/50
695/695 [==============================] - 78s 113ms/step - loss: 0.7936 - accuracy:
0.6838 - val_loss: 1.0710 - val_accuracy: 0.5946
Epoch 9/50
564/695 [=======================>......] - ETA: 14s - loss: 0.7727 - accuracy: 0.6890
```

```
from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(keras.layers.LSTM(16, dropout=0.8, return_sequences=True))
#model.add(keras.layers.LSTM(16, kernel_regularizer=l2(0.01), recurrent_regularizer=l2(0.01),
bias_regularizer=l2(0.01)))
model.add(keras.layers.LSTM(16, dropout=0.8))
#model.add(Dropout(0.50))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
#opt="adam"

model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
   x= x_train, y=y_train, batch_size=64, epochs=50, validation_data=(x_val, y_val))
```

```
695/695 [==============================] - 39s 55ms/step - loss: 0.8181 - accuracy:
0.6813 - val_loss: 0.9859 - val_accuracy: 0.5933
Epoch 47/50
695/695 [==============================] - 38s 55ms/step - loss: 0.8122 - accuracy:
0.6840 - val_loss: 1.0112 - val_accuracy: 0.5905
Epoch 48/50
695/695 [==============================] - 39s 56ms/step - loss: 0.8121 - accuracy:
0.6858 - val_loss: 1.0082 - val_accuracy: 0.5899
Epoch 49/50
695/695 [==============================] - 39s 56ms/step - loss: 0.8165 - accuracy:
0.6832 - val_loss: 1.0055 - val_accuracy: 0.5918
Epoch 50/50
695/695 [==============================] - 38s 55ms/step - loss: 0.8165 - accuracy:
0.6846 - val_loss: 1.0057 - val_accuracy: 0.5917
```





```
from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
```

```
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(keras.layers.LSTM(16, dropout=0.8, return_sequences=True))
#model.add(keras.layers.LSTM(16, kernel_regularizer=l2(0.01), recurrent_regularizer=l2(0.01),
bias_regularizer=l2(0.01)))
model.add(keras.layers.LSTM(16, dropout=0.8))
#model.add(Dropout(0.50))
model.add(keras.layers.Dense(5, activation='softmax'))
# opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
opt = tf.keras.optimizers.SGD(lr=0.2, clipnorm=1.)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
#opt="adam"

model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
  x= x_train, y=y_train, batch_size=64, epochs=50, validation_data=(x_val, y_val))
Epoch 1/50
695/695 [==============================] - 21s 30ms/step - loss: 1.3878 - accuracy: 0.4367
- val_loss: 1.2715 - val_accuracy: 0.5129
Epoch 2/50
695/695 [==============================] - 19s 28ms/step - loss: 1.2016 - accuracy: 0.5310
- val_loss: 1.1570 - val_accuracy: 0.5569
Epoch 3/50
695/695 [==============================] - 20s 28ms/step - loss: 1.0997 - accuracy: 0.5579
- val_loss: 1.1100 - val_accuracy: 0.5697
Epoch 4/50
695/695 [==============================] - 19s 28ms/step - loss: 0.9978 - accuracy: 0.5948
- val_loss: 1.1463 - val_accuracy: 0.5797
Epoch 5/50
695/695 [==============================] - 19s 28ms/step - loss: 0.9426 - accuracy: 0.6242
- val_loss: 1.0574 - val_accuracy: 0.5877
Epoch 6/50
695/695 [==============================] - 19s 28ms/step - loss: 0.9127 - accuracy: 0.6366
- val_loss: 1.0727 - val_accuracy: 0.5870
Epoch 7/50
695/695 [==============================] - 19s 28ms/step - loss: 0.8803 - accuracy: 0.6525
- val_loss: 1.0812 - val_accuracy: 0.5906
Epoch 8/50
695/695 [==============================] - 20s 28ms/step - loss: 0.8602 - accuracy: 0.6621
- val_loss: 1.0800 - val_accuracy: 0.5936
Epoch 9/50
695/695 [==============================] - 20s 28ms/step - loss: 0.8471 - accuracy: 0.6687
- val_loss: 1.0600 - val_accuracy: 0.5914
Epoch 10/50
```

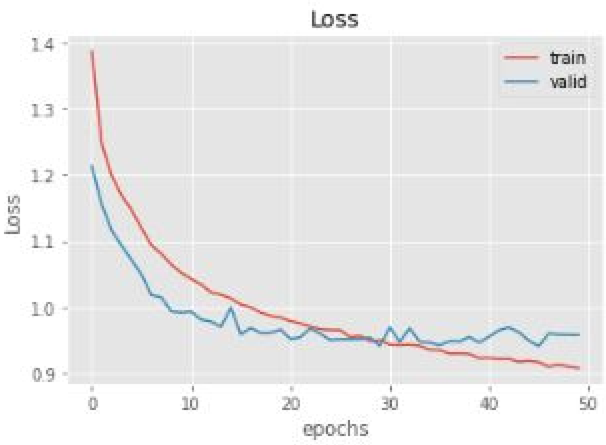
```
695/695 [==============================] - 19s 28ms/step - loss: 0.8362 - accuracy: 0.6738
- val_loss: 1.0839 - val_accuracy: 0.5861
Epoch 11/50
105/695 [===>..........................] - ETA: 15s - loss: 0.8505 - accuracy: 0.6679
```

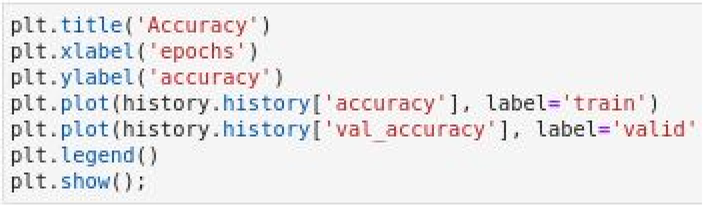
```
from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(keras.layers.LSTM(16, dropout=0.8, return_sequences=True))
#model.add(keras.layers.LSTM(16, kernel_regularizer=l2(0.01), recurrent_regularizer=l2(0.01),
bias_regularizer=l2(0.01)))
model.add(keras.layers.LSTM(16, dropout=0.8))
#model.add(Dropout(0.50))
model.add(keras.layers.Dense(5, activation='softmax'))
# opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
opt = tf.keras.optimizers.SGD(lr=0.1, decay=1e-6, momentum=0.9, nesterov=True)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
#opt="adam"

model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
   x= x_train, y=y_train, batch_size=64, epochs=50, validation_data=(x_val, y_val))
Epoch 30/50
695/695 [==============================] - 16s 23ms/step - loss: 0.7744 - accuracy:
0.6951 - val_loss: 1.0391 - val_accuracy: 0.5855
Epoch 31/50
695/695 [==============================] - 16s 23ms/step - loss: 0.7696 - accuracy:
0.6975 - val_loss: 1.0160 - val_accuracy: 0.5867
Epoch 32/50
695/695 [==============================] - 17s 24ms/step - loss: 0.7681 - accuracy:
0.6968 - val_loss: 1.0274 - val_accuracy: 0.5899
Epoch 33/50
695/695 [==============================] - 17s 24ms/step - loss: 0.7639 - accuracy:
0.7001 - val_loss: 1.0191 - val_accuracy: 0.5887
Epoch 34/50
695/695 [==============================] - 16s 23ms/step - loss: 0.7632 - accuracy:
0.7022 - val_loss: 1.0366 - val_accuracy: 0.5865
Epoch 35/50
```

```
695/695 [==============================] - 16s 24ms/step - loss: 0.7657 - accuracy:
0.7002 - val_loss: 1.0367 - val_accuracy: 0.5852
Epoch 36/50
695/695 [==============================] - 16s 24ms/step - loss: 0.7610 - accuracy:
0.7014 - val_loss: 1.0217 - val_accuracy: 0.5838
Epoch 37/50
695/695 [==============================] - 16s 24ms/step - loss: 0.7587 - accuracy:
0.7040 - val_loss: 1.0238 - val_accuracy: 0.5838
```

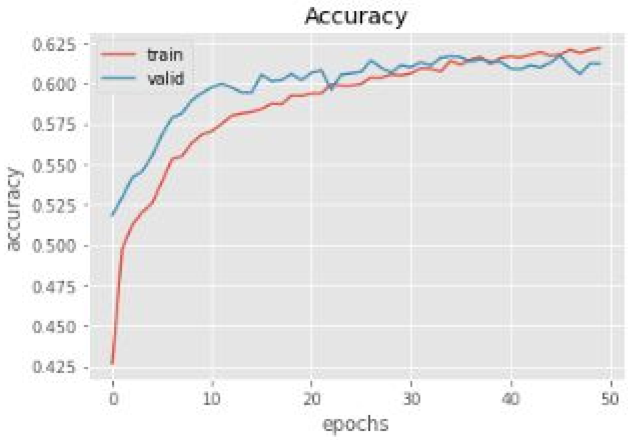
```
from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(keras.layers.LSTM(128, dropout=0.5))
#model.add(Dropout(0.50))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
   x= x_train, y=y_train, batch_size=16, epochs=50, validation_data=(x_val, y_val))
```

```
2779/2779 [==============================] - 54s 19ms/step - loss: 0.9167 - accuracy:
0.6178 - val_loss: 0.9409 - val_accuracy: 0.6170
Epoch 47/50
2779/2779 [==============================] - 54s 19ms/step - loss: 0.9098 - accuracy:
0.6206 - val_loss: 0.9599 - val_accuracy: 0.6105
Epoch 48/50
2779/2779 [==============================] - 54s 19ms/step - loss: 0.9133 - accuracy:
0.6187 - val_loss: 0.9585 - val_accuracy: 0.6056
Epoch 49/50
2779/2779 [==============================] - 54s 19ms/step - loss: 0.9098 - accuracy:
0.6206 - val_loss: 0.9580 - val_accuracy: 0.6121
Epoch 50/50
2779/2779 [==============================] - 54s 19ms/step - loss: 0.9076 - accuracy:
0.6216 - val_loss: 0.9578 - val_accuracy: 0.6120
```





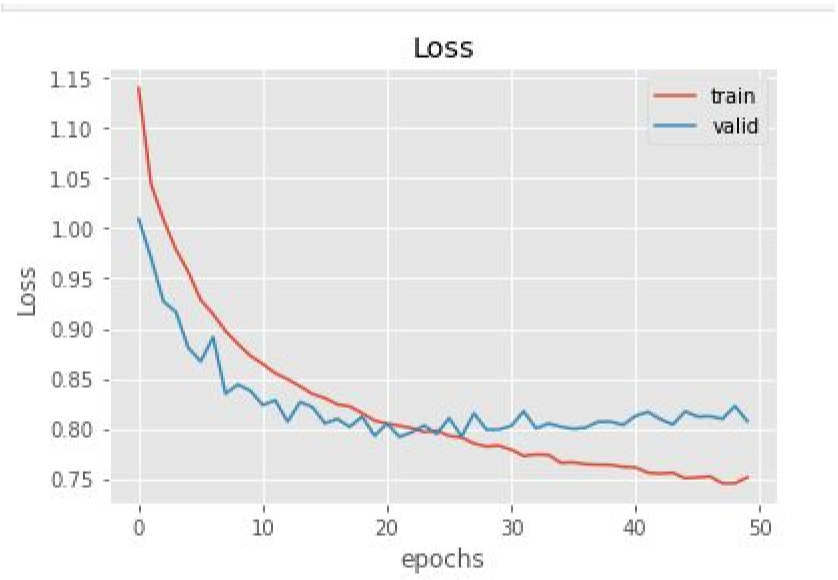
```
plt.title('Accuracy')
plt.xlabel('epochs')
plt.ylabel('accuracy')
plt.plot(history.history['accuracy'], label='train')
plt.plot(history.history['val_accuracy'], label='valid'
plt.legend()
plt.show();
```

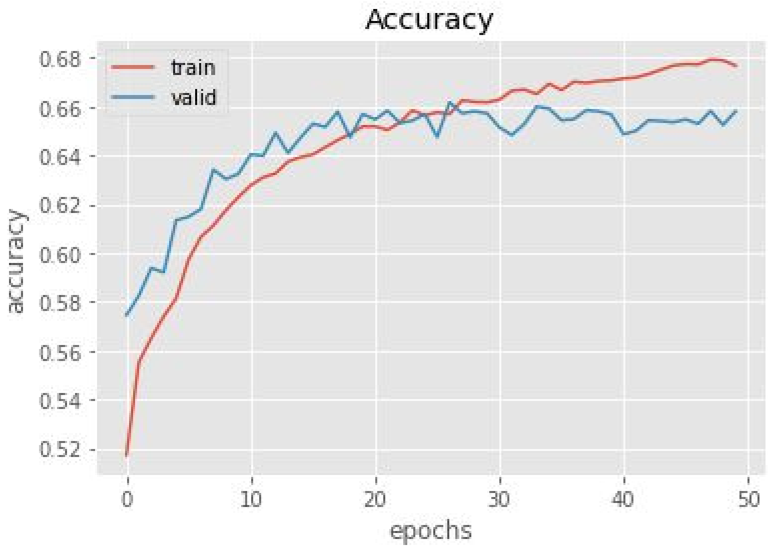


```
d = b2wCorpus.index[b2wCorpus["overall_rating"] == 2].tolist()
b2wCorpus=b2wCorpus.drop(b2wCorpus.index[d])
b2wCorpus['overall_rating'].replace({5: 2}, inplace = True)
print(b2wCorpus.head)

from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(keras.layers.LSTM(128, dropout=0.5))
#model.add(Dropout(0.50))
```

```
model.add(keras.layers.Dense(4, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
   x= x_train, y=y_train, batch_size=16, epochs=50, validation_data=(x_val, y_val))
```

```
Epoch 48/50
2547/2547 [==============================] - 47s 18ms/step - loss: 0.7455 - accuracy:
0.6794 - val_loss: 0.8098 - val_accuracy: 0.6583
Epoch 49/50
2547/2547 [==============================] - 47s 18ms/step - loss: 0.7454 - accuracy:
0.6790 - val_loss: 0.8226 - val_accuracy: 0.6525
Epoch 50/50
2547/2547 [==============================] - 47s 18ms/step - loss: 0.7514 - accuracy:
0.6768 - val_loss: 0.8074 - val_accuracy: 0.6581
```

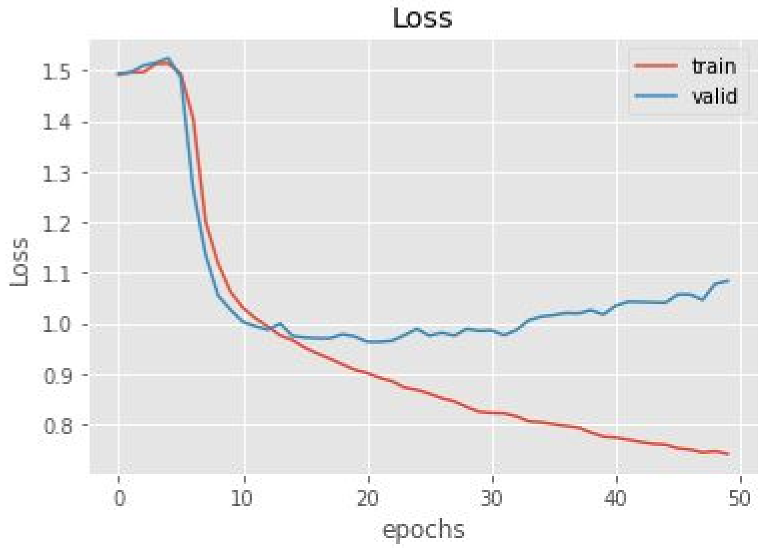


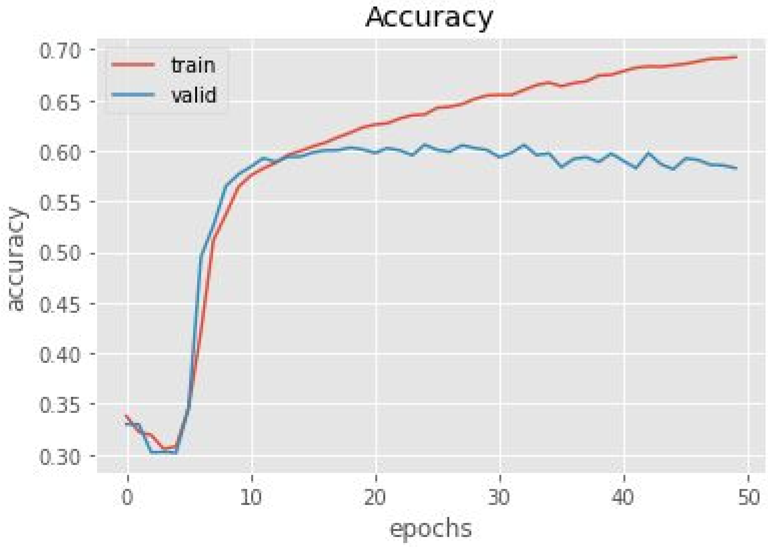


```
from tensorflow.keras.losses import sparse_categorical_crossentropy
SEQUENCE_MAXLEN = 50
model = keras.Sequential()
model.add(layers.Input(shape=(SEQUENCE_MAXLEN, )))
model.add(emb)
model.add(keras.layers.LSTM(128, dropout=0.25))
#model.add(Dropout(0.50))
model.add(keras.layers.Dense(5, activation='softmax'))
#opt = tf.keras.optimizers.SGD(learning_rate=.1, momentum=.5)
#opt = tf.keras.optimizers.Adamax(learning_rate=0.01, beta_1=.9, beta_2=.9, epsilon=1e-07,
name="Adamax")
#model.compile(optimizer=opt, loss=categorical_crossentropy, metrics=['acc', 'mae'])
opt="adam"
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=["accuracy"])
history = model.fit(
    x= x_train, y=y_train, batch_size=16, epochs=50, validation_data=(x_val, y_val))
```

```
Epoch 47/50
2779/2779 [==============================] - 60s 22ms/step - loss: 0.7502 - accuracy:
0.6882 - val_loss: 1.0569 - val_accuracy: 0.5908
Epoch 48/50
2779/2779 [==============================] - 60s 21ms/step - loss: 0.7446 - accuracy:
0.6907 - val_loss: 1.0466 - val_accuracy: 0.5860
Epoch 49/50
2779/2779 [==============================] - 60s 22ms/step - loss: 0.7469 - accuracy:
0.6912 - val_loss: 1.0782 - val_accuracy: 0.5855
Epoch 50/50
```

```
2779/2779 [==============================] - 61s 22ms/step - loss: 0.7418 - accuracy:
0.6925 - val_loss: 1.0839 - val_accuracy: 0.5827
```


Loss
train
valid
Loss
epochs


Accuracy
train
valid
accuracy
epochs