

```
!pip install emoji

Requirement already satisfied: emoji in /usr/local/lib/python3.10/dist-packages (2.6.0)

import emoji
import numpy as np
import pandas as pd

mapping = pd.read_csv("/content/sample_data/Mapping.csv")
output = pd.read_csv("/content/sample_data/OutputFormat.csv")
train = pd.read_csv("/content/sample_data/Train.csv")
test = pd.read_csv("/content/sample_data/Test.csv")
mapping
```

	Unnamed: 0	emoticons	number	
0	0	😬	0	
1	1	📷	1	
2	2	😬	2	
3	3	😬	3	
4	4	😬	4	
5	5	🌲	5	
6	6		6	
7	7	🍌	7	
8	8	😬	8	
9	9	❤️	9	
10	10	😬	10	
11	11	us	11	
12	12	☀️	12	
13	13	✨	13	
14	14	💙	14	
15	15	💕	15	
16	16	😬	16	
17	17	😬	17	
18	18	💜	18	
19	19	💯	19	

```
mapping = mapping.drop("Unnamed: 0", axis=1)
mapping.head()
```

```
mapping = mapping.set_index('number')
mapping.head()
```

emoticons	
number	
0	😏
1	📷
2	😘
3	😂
4	😊

```
mapping_dict = mapping.to_dict()
mapping_dict = mapping_dict['emoticons']
mapping_dict
```

```
{0: '😏',
1: '📷',
2: '😘',
3: '😂',
4: '😊',
...}
```

Saved successfully!

```
8: '😏',
9: '❤️',
10: '😏',
11: 'us',
12: '☀️',
13: '✨',
14: '💙',
15: '💕',
16: '😎',
17: '😊',
18: '💜',
19: '💯'}
```

```
train.head()
```

Unnamed: 0		TEXT	Label
0	0	Vacation wasted ! #vacation2017 #photobomb #ti...	0
1	1	Oh Wynwood, you're so funny! : @user #Wynwood ...	1
2	2	Been friends since 7th grade. Look at us now w...	2
3	3	This is what it looks like when someone loves ...	3
4	4	RT @user this white family was invited to a Bl...	3

```
train.columns

Index(['Unnamed: 0', 'TEXT', 'Label'], dtype='object')

train = train.drop('Unnamed: 0', axis=1)

train.tail()
```

	TEXT	Label	
69995	Yes, I call Galina "my Bubie" Go follow my bea...	3	
69996	I SEA you, Seattle @ Ballard Seafood Festival\n	16	
69997	If one of my daughters is wearing this and ask...	2	
69998	Guess who whoop people on THEIR homecoming?! #...	3	
69999	We Love you Robbie @ Heritage Memorial Cemeter...	14	

```
train.head()
```

	TEXT	Label	
0	Vacation wasted ! #vacation2017 #photobomb #ti...	0	
1	Oh Wynwood, you're so funny! : @user #Wynwood ...	1	
2	Been friends since 7th grade. Look at us now w...	2	
3	This is what it looks like when someone loves ...	3	
4	RT @user this white family was invited to a Bl...	3	

```
X_train = train['TEXT'].values
y_train = train['Label'].values
```

Saved successfully!

```
file = open("/content/sample_data/glove.6B.50d.txt", encoding = 'utf8')
```

```
file
```

```
<_io.TextIOWrapper name='/content/sample_data/glove.6B.50d.txt' mode='r' encoding='utf8'>
```

```
def intialize_emb_matrix(file):
    embedding_matrix = {}
    for line in file:
        values = line.split()
        word = values[0]
        embedding = np.array(values[1:], dtype='float64')
        embedding_matrix[word] = embedding

    return embedding_matrix
```

```
embedding_matrix = intialize_emb_matrix(file)
```

```
embedding_matrix['dance']
```

```
array([-9.9716e-01,  6.1348e-01, -1.7686e+00,  8.7052e-02, -2.4644e-01,
        1.9984e-01, -2.5892e-01, -3.1309e-02, -4.3117e-01,  1.0844e+00,
        4.7032e-01, -3.6731e-01,  1.0047e-01,  1.2459e+00,  2.5728e-03,
       -3.8805e-01,  4.8489e-01,  3.8052e-01, -5.6983e-01, -8.4332e-01,
        8.8607e-01,  9.7396e-01,  9.0600e-03,  9.5187e-01,  3.3867e-01,
       -2.5218e-01, -1.5633e+00, -4.8921e-01, -3.4712e-01, -1.2629e+00,
        2.8455e+00,  2.8887e-01, -1.9158e-01,  6.5195e-02, -1.9799e-01,
        2.2815e-01,  4.4613e-01, -8.2659e-02, -3.3068e-01,  4.8144e-02,
       -2.7224e-01, -2.1765e-01, -5.5931e-01, -4.0149e-01,  5.6395e-01,
       -1.9915e-01,  7.2645e-01, -1.5184e+00, -8.7863e-01,  2.1860e-01])
```



```
[ -1.0919e-05,  3.3524e-01,  3.5745e-01, ..., -4.5097e-01,
  -4.8969e-02,  1.1316e+00],
...,
[ 0.0000e+00,  0.0000e+00,  0.0000e+00, ...,  0.0000e+00,
  0.0000e+00,  0.0000e+00],
[ 0.0000e+00,  0.0000e+00,  0.0000e+00, ...,  0.0000e+00,
  0.0000e+00,  0.0000e+00],
[ 0.0000e+00,  0.0000e+00,  0.0000e+00, ...,  0.0000e+00,
  0.0000e+00,  0.0000e+00]]])
```

X\_temb.shape

```
(70000, 168, 50)
```

```
from keras.utils.np_utils import to_categorical
```

```
y_train = to_categorical(y_train)
```

y\_train

```
array([[1., 0., 0., ..., 0., 0., 0.],
       [0., 1., 0., ..., 0., 0., 0.],
       [0., 0., 1., ..., 0., 0., 0.],
       ...,
       [0., 0., 1., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]], dtype=float32)
```

Saved successfully!



```
from keras.models import Sequential
from keras.layers import LSTM, SimpleRNN, Dense, Dropout
```

```
model = Sequential()
```

```
lm = len(mapping_dict)
lm
```

```
20
```

```
model.add(LSTM(units = 256, return_sequences=True, input_shape = (168,50)))
model.add(Dropout(0.3))
model.add(LSTM(units=128))
model.add(Dropout(0.3))
model.add(Dense(units=128, activation='relu'))
model.add(Dense(units=64, activation='relu'))
model.add(Dense(units=32, activation='relu'))
model.add(Dense(units=20, activation='relu'))
model.add(Dense(units=20, activation='softmax'))
```

```
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
lstm (LSTM)	(None, 168, 256)	314368
dropout (Dropout)	(None, 168, 256)	0
lstm_1 (LSTM)	(None, 128)	197120
dropout_1 (Dropout)	(None, 128)	0

7/4/23, 10:21 PM

Emoji\_prediction.ipynb - Colaboratory

dense (Dense)

(None, 128)

16512

dense\_1 (Dense)

(None, 64)

8256

dense\_2 (Dense)

(None, 32)

2080

dense\_3 (Dense)

(None, 20)

660

dense\_4 (Dense)

(None, 20)

420

=====

Total params: 539,416

Trainable params: 539,416

Non-trainable params: 0

```
model.compile(optimizer='adam', loss=keras.losses.categorical_crossentropy, metrics=['acc'])
```

```
res = model.fit(X_temb, y_train, validation_split=0.2, batch_size=32, epochs=10, verbose=2)
```

```
Epoch 1/10
1750/1750 - 46s - loss: 2.7455 - acc: 0.2146 - val_loss: 2.7384 - val_acc: 0.2183 - 46s/epoch - 26ms/st
Epoch 2/10
1750/1750 - 30s - loss: 2.7386 - acc: 0.2149 - val_loss: 2.7366 - val_acc: 0.2183 - 30s/epoch - 17ms/st
Epoch 3/10
1750/1750 - 34s - loss: 2.7376 - acc: 0.2149 - val_loss: 2.7370 - val_acc: 0.2183 - 34s/epoch - 19ms/st
Epoch 4/10
1750/1750 - 31s - loss: 2.7368 - acc: 0.2149 - val_loss: 2.7361 - val_acc: 0.2183 - 31s/epoch - 18ms/st
Epoch 5/10
1750/1750 - 30s - loss: 2.7363 - acc: 0.2149 - val_loss: 2.7363 - val_acc: 0.2183 - 30s/epoch - 17ms/st
Epoch 6/10
1750/1750 - 31s - loss: 2.7369 - acc: 0.2149 - val_loss: 2.7369 - val_acc: 0.2183 - 31s/epoch - 18ms/st
Epoch 7/10
1750/1750 - 31s - loss: 2.7364 - acc: 0.2149 - val_loss: 2.7362 - val_acc: 0.2183 - 31s/epoch - 18ms/st
Epoch 8/10
1750/1750 - 33s - loss: 2.7372 - acc: 0.2149 - val_loss: 2.7348 - val_acc: 0.2183 - 33s/epoch - 19ms/st
Epoch 9/10
1750/1750 - 32s - loss: 2.7354 - acc: 0.2149 - val_loss: 2.7344 - val_acc: 0.2183 - 32s/epoch - 19ms/st
Epoch 10/10
1750/1750 - 31s - loss: 2.7338 - acc: 0.2149 - val_loss: 2.7335 - val_acc: 0.2183 - 31s/epoch - 17ms/st
```

Saved successfully!

```
train_data = pd.read_csv("/content/sample_data/train_emoji.csv", header=None)
test_data = pd.read_csv("/content/sample_data/test_emoji.csv", header=None)
```

```
train_data.head()
```

		0	1	2	3	
0	never talk to me again	3	NaN	NaN		
1	I am proud of your achievements	2	NaN	NaN		
2	It is the worst day in my life	3	NaN	NaN		
3	Miss you so much	0	NaN	[0]		
4	food is life	4	NaN	NaN		

```
test_data.head()
```

	0	1	
0	I want to eat	4	
1	he did not answer	3	

```
train_data.drop(labels=[2, 3], axis=1, inplace=True)
```

```
3 she got me a nice present
```

```
train_data.tail()
```

	0	1	
127	he had to make a home run	1	
128	I am ordering food	4	
129	What is wrong with you	3	
130	I love you	0	
131	great job	2	

```
import emoji
```

```
pip install emoji==1.7
```

Saved successfully!



gz (175 kB)

175.4/175.4 kB 5.7 MB/s eta 0:00:00

```
Preparing metadata (setup.py) ... done
Building wheels for collected packages: emoji
Building wheel for emoji (setup.py) ... done
Created wheel for emoji: filename=emoji-1.7.0-py3-none-any.whl size=171033 sha256=f5394ef299407d719c7
Stored in directory: /root/.cache/pip/wheels/31/8a/8c/315c9e5d7773f74b33d5ed33f075b49c6eaeb7cedbb86e2
Successfully built emoji
Installing collected packages: emoji
Attempting uninstall: emoji
Found existing installation: emoji 2.6.0
Uninstalling emoji-2.6.0:
Successfully uninstalled emoji-2.6.0
Successfully installed emoji-1.7.0
WARNING: The following packages were previously imported in this runtime:
[emoji]
You must restart the runtime in order to use newly installed versions.
```

RESTART RUNTIME

```
def extract_emojis(s):
    return ''.join(c for c in s if c in emoji.UNICODE_EMOJI['en'])
```

```
emoji.emojize(':angry_face:')
```

```
'🙄'
```

```
emoji_mapping = {
```

```
    '0': ':beating_heart:',
```

```
    '1': ':baseball:',
```

```
    '2': ':beaming_face_with_smiling_eyes:',
```

```
    '3': ':angry_face:',
```

```
'4': ':face_savoring_food:'  
  
}  
  
for key, value in emoji_mapping.items():  
    emoji_mapping[key] = emoji.emojize(value)  
  
print(emoji_mapping)  
  
{'0': '❤️', '1': '⚾️', '2': '😄', '3': '😡', '4': '😋'}  
  
X_train = train_data[0].values  
y_train = train_data[1].values  
  
file  
  
<_io.TextIOWrapper name='/content/sample_data/glove.6B.50d.txt' mode='r' encoding='utf8'>  
  
embedding_matrix
```

Saved successfully!





```
attention : array([-0.084238,  0.53055,  0.12816, -0.28886, -0.18125,
 -0.205, -0.096976,  0.070474,  0.23336,  0.14649,
 -0.55293, -0.022887, -0.35714, -0.35338,  0.61158,
 -0.21432, -0.024156, -0.49679,  0.044917, -0.55818,
  0.32167,  0.56808,  0.18367, -0.0069878,  0.7587,
 -1.6715, -0.61596, -0.35291,  0.11358,  0.53997,
  2.8148,  0.85088,  0.12374, -1.0855, -0.49705,
  0.35443, -0.43236, -0.181, -0.37668, -0.58063,
  0.28078,  0.63339, -0.046038, -0.091437,  0.59663,
  0.066547,  0.062638,  0.5009,  0.23662,  0.16071 ]),
...}
```

```
X_train = get_emb_data(X_train, 10)
```

```
X_train
```

```
[[ 6.1183e-01, -2.2072e-01, -1.0898e-01, ..., -4.3688e-02,
 -9.7922e-02,  1.6806e-01],
 [ 6.1850e-01,  6.4254e-01, -4.6552e-01, ..., -2.7557e-01,
  3.0899e-01,  4.8497e-01],
 [ 4.1800e-01,  2.4968e-01, -4.1242e-01, ..., -1.8411e-01,
 -1.1514e-01, -7.8581e-01],
 ...,
 [ 5.1491e-01,  8.8806e-01, -7.1906e-01, ..., -9.9837e-01,
 -6.6328e-02,  1.3118e-01],
 [ 0.0000e+00,  0.0000e+00,  0.0000e+00, ...,  0.0000e+00,
  0.0000e+00,  0.0000e+00],
 [ 0.0000e+00,  0.0000e+00,  0.0000e+00, ...,  0.0000e+00,
  0.0000e+00,  0.0000e+00]],
```

Saved successfully!



```
[[ 4.5323e-01,  5.9811e-02, -1.0577e-01, ...,  5.3240e-01,
 -2.5103e-01,  6.2546e-01],
 [ 6.1850e-01,  6.4254e-01, -4.6552e-01, ..., -2.7557e-01,
  3.0899e-01,  4.8497e-01],
 [ 1.3423e-01, -7.0572e-01,  2.5652e-01, ...,  4.3723e-01,
  9.0636e-01,  5.1193e-01],
 ...,
 [ 0.0000e+00,  0.0000e+00,  0.0000e+00, ...,  0.0000e+00,
  0.0000e+00,  0.0000e+00],
 [ 0.0000e+00,  0.0000e+00,  0.0000e+00, ...,  0.0000e+00,
  0.0000e+00,  0.0000e+00]],
```

```
0.0000e+00, 0.0000e+00],
[ 0.0000e+00, 0.0000e+00, 0.0000e+00, ..., 0.0000e+00,
 0.0000e+00, 0.0000e+00]])
```

```
y_train = to_categorical(y_train)
```

y\_train

```
[0., 0., 1., 0., 0.],
[0., 0., 1., 0., 0.],
[1., 0., 0., 0., 0.],
[1., 0., 0., 0., 0.],
[0., 0., 0., 1., 0.],
[0., 1., 0., 0., 0.],
[0., 0., 1., 0., 0.],
[0., 1., 0., 0., 0.],
[0., 0., 1., 0., 0.],
[0., 0., 1., 0., 0.],
[0., 0., 0., 0., 1.],
[0., 0., 0., 1., 0.],
[0., 0., 0., 1., 0.],
[0., 0., 1., 0., 0.],
[0., 0., 0., 0., 1.],
[1., 0., 0., 0., 0.],
[1., 0., 0., 0., 0.],
[0., 0., 0., 1., 0.],
[0., 0., 0., 1., 0.],
[0., 0., 0., 1., 0.],
[0., 0., 1., 0., 0.],
[1., 0., 0., 0., 0.]
```

Saved successfully!

```
[0., 0., 0., 1., 0.],
[1., 0., 0., 0., 0.],
[0., 0., 1., 0., 0.],
[0., 0., 1., 0., 0.],
[0., 0., 1., 0., 0.],
[0., 0., 0., 1., 0.],
[0., 0., 1., 0., 0.],
[0., 0., 1., 0., 0.],
[0., 0., 1., 0., 0.],
[0., 0., 0., 0., 1.],
[0., 1., 0., 0., 0.],
[0., 1., 0., 0., 0.],
[0., 0., 0., 1., 0.],
[0., 0., 0., 1., 0.],
[0., 0., 0., 0., 1.],
[0., 1., 0., 0., 0.],
[0., 0., 1., 0., 0.],
[0., 1., 0., 0., 0.],
[0., 1., 0., 0., 0.],
[0., 0., 0., 1., 0.],
[0., 1., 0., 0., 0.],
[1., 0., 0., 0., 0.],
[0., 0., 0., 0., 1.],
[1., 0., 0., 0., 0.],
[0., 0., 0., 1., 0.],
[0., 0., 0., 1., 0.],
[0., 0., 0., 0., 1.],
[0., 0., 0., 0., 1.],
[0., 1., 0., 0., 0.],
[0., 0., 0., 0., 1.],
[0., 0., 0., 1., 0.],
[1., 0., 0., 0., 0.],
[0., 0., 1., 0., 0.]], dtype=float32)
```

```
model_tte = Sequential()
```

```
model_tte.add(LSTM(units=64, input_shape=(10, 50), return_sequences=True))
```

```
model_tte.add(Dropout(0.3))

model_tte.add(LSTM(units=32))

model_tte.add(Dropout(0.2))

model_tte.add(Dense(units=10, activation='relu'))

model_tte.add(Dense(units=5, activation='softmax'))

model_tte.summary()
```

Model: "sequential\_1"

Layer (type)	Output Shape	Param #
lstm_2 (LSTM)	(None, 10, 64)	29440
dropout_2 (Dropout)	(None, 10, 64)	0
lstm_3 (LSTM)	(None, 32)	12416
dropout_3 (Dropout)	(None, 32)	0
dense_5 (Dense)	(None, 10)	330
dense_6 (Dense)	(None, 5)	55

Saved successfully!

```
model_tte.compile(optimizer='adam', loss=keras.losses.categorical_crossentropy, metrics=['acc'])

history = model_tte.fit(X_train, y_train, validation_split=0.1, verbose=2, batch_size=32, epochs=100)
```

7/4/23, 10:21 PM

Emoji\_prediction.ipynb - Colaboratory

4/4 - 0s - loss: 0.0619 - acc: 0.9746 - val\_loss: 1.2596 - val\_acc: 0.6429 - 49ms/epoch - 12ms/step  
Epoch 89/100  
4/4 - 0s - loss: 0.0602 - acc: 0.9831 - val\_loss: 1.1608 - val\_acc: 0.7857 - 50ms/epoch - 13ms/step  
Epoch 90/100  
4/4 - 0s - loss: 0.0611 - acc: 0.9831 - val\_loss: 1.2588 - val\_acc: 0.7143 - 53ms/epoch - 13ms/step  
Epoch 91/100  
4/4 - 0s - loss: 0.0603 - acc: 0.9746 - val\_loss: 1.3976 - val\_acc: 0.7143 - 49ms/epoch - 12ms/step  
Epoch 92/100  
4/4 - 0s - loss: 0.0628 - acc: 0.9831 - val\_loss: 1.3440 - val\_acc: 0.7857 - 48ms/epoch - 12ms/step  
Epoch 93/100  
4/4 - 0s - loss: 0.0585 - acc: 0.9831 - val\_loss: 1.3269 - val\_acc: 0.7857 - 49ms/epoch - 12ms/step  
Epoch 94/100  
4/4 - 0s - loss: 0.0611 - acc: 0.9746 - val\_loss: 1.2100 - val\_acc: 0.7857 - 49ms/epoch - 12ms/step  
Epoch 95/100  
4/4 - 0s - loss: 0.0513 - acc: 0.9831 - val\_loss: 1.4465 - val\_acc: 0.6429 - 50ms/epoch - 12ms/step  
Epoch 96/100  
4/4 - 0s - loss: 0.0650 - acc: 0.9746 - val\_loss: 1.7042 - val\_acc: 0.6429 - 49ms/epoch - 12ms/step  
Epoch 97/100  
4/4 - 0s - loss: 0.0488 - acc: 0.9831 - val\_loss: 1.6446 - val\_acc: 0.6429 - 47ms/epoch - 12ms/step  
Epoch 98/100  
4/4 - 0s - loss: 0.0574 - acc: 0.9746 - val\_loss: 1.5801 - val\_acc: 0.6429 - 48ms/epoch - 12ms/step  
Epoch 99/100  
4/4 - 0s - loss: 0.0519 - acc: 0.9831 - val\_loss: 1.5287 - val\_acc: 0.6429 - 53ms/epoch - 13ms/step  
Epoch 100/100  
4/4 - 0s - loss: 0.0447 - acc: 0.9831 - val\_loss: 1.6838 - val\_acc: 0.6429 - 50ms/epoch - 13ms/step

```
model_tte2 = Sequential()  
model_tte2.add(LSTM(input_shape=(10, 50), return_sequences=True))
```

Saved successfully!

```
model_tte2.add(LSTM(units=64))  
model_tte2.add(Dropout(0.2))  
model_tte2.add(Dense(units=32, activation='relu'))  
model_tte2.add(Dense(units=20, activation='relu'))  
model_tte2.add(Dense(units=5, activation='relu'))  
  
model_tte2.summary()
```

Model: "sequential\_2"

Layer (type)	Output Shape	Param #
=====		
lstm_4 (LSTM)	(None, 10, 128)	91648
dropout_4 (Dropout)	(None, 10, 128)	0
lstm_5 (LSTM)	(None, 64)	49408
dropout_5 (Dropout)	(None, 64)	0
dense_7 (Dense)	(None, 32)	2080
dense_8 (Dense)	(None, 20)	660
dense_9 (Dense)	(None, 5)	105
=====		
Total params: 143,901		
Trainable params: 143,901		
Non-trainable params: 0		

```
model_tte2.compile(optimizer='adam', loss=keras.losses.categorical_crossentropy, metrics=['acc'])
```

```
history2 = model_tte2.fit(X_train, y_train, validation_split=0.1, verbose=2, batch_size=32, epochs=100)
```

```
Epoch 72/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 51ms/epoch - 13ms/step
Epoch 73/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 50ms/epoch - 13ms/step
Epoch 74/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 51ms/epoch - 13ms/step
Epoch 75/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 58ms/epoch - 15ms/step
Epoch 76/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 68ms/epoch - 17ms/step
Epoch 77/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 51ms/epoch - 13ms/step
Epoch 78/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 71ms/epoch - 18ms/step
Epoch 79/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 51ms/epoch - 13ms/step
Epoch 80/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 51ms/epoch - 13ms/step
Epoch 81/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 51ms/epoch - 13ms/step
Epoch 82/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 50ms/epoch - 13ms/step
Epoch 83/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 51ms/epoch - 13ms/step
Epoch 84/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 52ms/epoch - 13ms/step
Epoch 85/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 51ms/epoch - 13ms/step
Epoch 86/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 55ms/epoch - 14ms/step
Epoch 87/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 50ms/epoch - 13ms/step
Epoch 88/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 51ms/epoch - 13ms/step
Epoch 89/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 50ms/epoch - 13ms/step
Epoch 90/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 56ms/epoch - 14ms/step
Epoch 91/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 50ms/epoch - 13ms/step
Epoch 92/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 73ms/epoch - 18ms/step
Epoch 93/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 67ms/epoch - 17ms/step
Epoch 94/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 52ms/epoch - 13ms/step
Epoch 95/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 66ms/epoch - 17ms/step
Epoch 96/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 51ms/epoch - 13ms/step
Epoch 97/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 64ms/epoch - 16ms/step
Epoch 98/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 52ms/epoch - 13ms/step
Epoch 99/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 52ms/epoch - 13ms/step
Epoch 100/100
4/4 - 0s - loss: nan - acc: 0.1610 - val_loss: nan - val_acc: 0.2143 - 53ms/epoch - 13ms/step
```

Saved successfully!

```
model_tte3 = Sequential()
```

```
model_tte3.add(LSTM(units=128, input_shape=(10, 50), return_sequences=True))
```

```
model_tte3.add(Dropout(0.3))
```

```
model_tte3.add(LSTM(units=64))
```

```
model_tte3.add(LSTM(units=64,

model_tte3.add(Dropout(0.2))

model_tte3.add(Dense(units=32, activation='relu'))

model_tte3.add(Dense(units=20, activation='relu'))

model_tte3.add(Dense(units=5, activation='softmax'))

model_tte3.summary()
```

Model: "sequential\_3"

Layer (type)	Output Shape	Param #
=====		
lstm_6 (LSTM)	(None, 10, 128)	91648
dropout_6 (Dropout)	(None, 10, 128)	0
lstm_7 (LSTM)	(None, 64)	49408
dropout_7 (Dropout)	(None, 64)	0
dense_10 (Dense)	(None, 32)	2080
dense_11 (Dense)	(None, 20)	660
dense_12 (Dense)	(None, 5)	105

Saved successfully!

Trainable params: 143,901  
Non-trainable params: 0

```
model_tte3.compile(optimizer='adam', loss=keras.losses.categorical_crossentropy, metrics=['acc'])

history3 = model_tte3.fit(X_train, y_train, validation_split=0.1, verbose=2, batch_size=32, epochs=100)
```

```

4/4 - 0s - loss: 0.0465 - acc: 0.9831 - val_loss: 2.9066 - val_acc: 0.4286 - 52ms/epoch - 13ms/step
Epoch 87/100
4/4 - 0s - loss: 0.0538 - acc: 0.9831 - val_loss: 3.2304 - val_acc: 0.4286 - 52ms/epoch - 13ms/step
Epoch 88/100
4/4 - 0s - loss: 0.0558 - acc: 0.9746 - val_loss: 3.2886 - val_acc: 0.4286 - 81ms/epoch - 20ms/step
Epoch 89/100
4/4 - 0s - loss: 0.0664 - acc: 0.9746 - val_loss: 2.1654 - val_acc: 0.5714 - 51ms/epoch - 13ms/step
Epoch 90/100
4/4 - 0s - loss: 0.0649 - acc: 0.9746 - val_loss: 2.0759 - val_acc: 0.6429 - 51ms/epoch - 13ms/step
Epoch 91/100
4/4 - 0s - loss: 0.1173 - acc: 0.9576 - val_loss: 3.5043 - val_acc: 0.4286 - 56ms/epoch - 14ms/step
Epoch 92/100
4/4 - 0s - loss: 0.0471 - acc: 0.9746 - val_loss: 2.9432 - val_acc: 0.4286 - 51ms/epoch - 13ms/step
Epoch 93/100
4/4 - 0s - loss: 0.0572 - acc: 0.9746 - val_loss: 2.1659 - val_acc: 0.5000 - 50ms/epoch - 13ms/step
Epoch 94/100
4/4 - 0s - loss: 0.0774 - acc: 0.9661 - val_loss: 2.8551 - val_acc: 0.5714 - 64ms/epoch - 16ms/step
Epoch 95/100
4/4 - 0s - loss: 0.1934 - acc: 0.9576 - val_loss: 1.9692 - val_acc: 0.4286 - 59ms/epoch - 15ms/step
Epoch 96/100
4/4 - 0s - loss: 0.0867 - acc: 0.9661 - val_loss: 2.5833 - val_acc: 0.6429 - 68ms/epoch - 17ms/step
Epoch 97/100
4/4 - 0s - loss: 0.1518 - acc: 0.9237 - val_loss: 3.7219 - val_acc: 0.4286 - 52ms/epoch - 13ms/step
Epoch 98/100
4/4 - 0s - loss: 0.3407 - acc: 0.9153 - val_loss: 1.8120 - val_acc: 0.5714 - 52ms/epoch - 13ms/step
Epoch 99/100
4/4 - 0s - loss: 0.1854 - acc: 0.9492 - val_loss: 1.8742 - val_acc: 0.7143 - 56ms/epoch - 14ms/step
Epoch 100/100
4/4 - 0s - loss: 0.1467 - acc: 0.9322 - val_loss: 2.6315 - val_acc: 0.4286 - 55ms/epoch - 14ms/step

```

```
test_data = nd.read_csv("/content/sample_data/test_emoji.csv", header=None)
```

Saved successfully!

```
test_data.head()
```

	0	1	
0	I want to eat!	4	
1	he did not answer!	3	
2	he got a very nice raise!	2	
3	she got me a nice present!	2	
4	ha ha ha it was so funny!	2	

```

test_data[0] = test_data[0].apply(lambda x:x[:-1])
X_test = test_data[0].values
y_test = test_data[1].values

```

```

X_test = get_emb_data(X_test, 10)
y_test = to_categorical(y_test)

```

```
model_tte3.evaluate(X_test, y_test)[1]*100
```

```

2/2 [=====] - 0s 9ms/step - loss: 2.4689 - acc: 0.5714
57.14285969734192

```

```
predicted = model_tte3.predict(X_test)
```

```
2/2 [=====] - 1s 8ms/step
```

```
classes = np.argmax(predicted,axis=1)
```

```
classes
```

```
array([4, 3, 2, 0, 2, 2, 3, 4, 4, 3, 2, 2, 2, 3, 1, 2, 2, 0, 3, 4, 3, 3,  
       2, 0, 3, 3, 3, 2, 3, 2, 0, 1, 2, 2, 3, 3, 0, 4, 4, 2, 4, 0, 0, 1,  
       2, 0, 3, 2, 3, 3, 3, 2, 3, 2, 2, 3])
```

```
for t in range(len(test)):  
    print(test_data[0].iloc[t])  
    print("Predictions:", emoji.emojize(emoji_mapping[str(classes[t])]))
```

Saved successfully!





I want to eat  
Predictions: 😊  
he did not answer  
Predictions: 😡  
he got a very nice raise  
Predictions: 😊  
she got me a nice present  
Predictions: ❤️  
ha ha ha it was so funny  
Predictions: 😊  
he is a good friend  
Predictions: 😊  
I am upset  
Predictions: 😡  
We had such a lovely dinner tonight  
Predictions: 😊  
where is the food  
Predictions: 😊  
Stop making this joke ha ha ha  
Predictions: 😡  
where is the ball  
Predictions: 😊  
work is hard  
Predictions: 😊  
This girl is messing with me  
Predictions: 😊  
are you serious  
Predictions: 😡  
Let us go play baseball  
Predictions: ⚾

Saved successfully!



Predictions: 😊  
Congratulation for having a baby  
Predictions: ❤️  
stop pissing me of  
Predictions: 😡  
any suggestions for dinner  
Predictions: 😊  
I love taking breaks  
Predictions: 😡  
you brighten my day  
Predictions: 😡  
I boiled rice  
Predictions: 😊  
she is a bully  
Predictions: ❤️  
Why are you feeling bad  
Predictions: 😡  
I am upset  
Predictions: 😡  
give me the ball  
Predictions: 😡  
My grandmother is the love of my life  
Predictions: 😊  
enjoy your game  
Predictions: 😡  
valentine day is near  
Predictions: 😊  
I miss you so much  
Predictions: ❤️  
throw the ball  
Predictions: ⚾  
My life is so boring  
Predictions: 😊  
she said yes  
Predictions: 😊  
will you be my valentine  
Predictions: 😡  
he can pitch really well  
Predictions: 😡

dance with me  
Predictions: 💕  
I am hungr  
Predictions: 😋  
See you at the restaurant  
Predictions: 😊  
I like to laugh  
Predictions: 😄  
I will ru  
Predictions: 😊  
I like your jacket  
Predictions: 💕  
i miss her  
Predictions: 💕  
what is your favorite baseball game  
Predictions: ⚾  
Good job  
Predictions: 😊  
I love you to the stars and back  
Predictions: 💕  
What you did was awesome  
Predictions: 😏  
ha ha ha lol  
Predictions: 😄

! 0s completed at 10:17 PM



Saved successfully!

