Name- Gaurav Sharan

DATASET LINK - https://www.kaggle.com/c/quora-question-pairs/data

Double-click (or enter) to edit

```
train=pd.read_csv('/kaggle/input/quora-question-pairs/train.csv.zip')
```

```
train.shape
```

```
(404290, 6)
```

We have 404290 observations and 6 features in the dataset.

```
train.head()
```

	id	qid1	qid2	question1	question2	is_duplicate
0	0	1	2	What is the step by step guide to invest in sh...	What is the step by step guide to invest in sh...	0
1	1	3	4	What is the story of Kohinoor (Koh-i-Noor) Dia...	What would happen if the Indian government sto...	0
2	2	5	6	How can I increase the speed of my internet co...	How can Internet speed be increased by hacking...	0
3	3	7	8	Why am I mentally very	Find the remainder when	0

```
test=pd.read_csv('/kaggle/input/quora-question-pairs/test.csv')
```

```
test.head()
```

	test_id	question1	question2
0	0	How does the Surface Pro himself 4 compare wit...	Why did Microsoft choose core m3 and not core ...
1	1	Should I have a hair transplant at age 24? How...	How much cost does hair transplant require?
2	2	What but is the best way to send money from Ch...	What you send money to China?
3	3	Which food not emulsifiers?	What foods fibre?
4	4	How "aberystwyth" start reading?	How their can I start reading?

```
train[train['is_duplicate']==1].head()
```

	id	qid1	qid2	question1	question2	is_duplicate
5	5	11	12	Astrology: I am a Capricorn Sun Cap moon and c...	I'm a triple Capricorn (Sun, Moon and ascendan...	1
7	7	15	16	How can I be a good geologist?	What should I do to be a great geologist?	1
11	11	23	24	How do I read and find my YouTube comments?	How can I see all my Youtube comments?	1
12	12	25	26	What can make Physics easy to learn?	How can you make physics easy to learn?	1
13	13	27	28	What was your first sexual experience like?	What was your first sexual experience?	1

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 404290 entries, 0 to 404289
Data columns (total 6 columns):
 #   Column        Non-Null Count   Dtype 
---  ------        --------------   ----- 
 0   id            404290 non-null  int64 
 1   qid1          404290 non-null  int64 
 2   qid2          404290 non-null  int64 
 3   question1     404289 non-null  object
 4   question2     404288 non-null  object
 5   is_duplicate  404290 non-null  int64 
dtypes: int64(4), object(2)
memory usage: 18.5+ MB
```

```
#dropping null values
train=train.dropna()
```

```
train_list1=list(train['question1'])
train_list2=list(train['question2'])
train_list=train_list1+train_list2
```

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
```

```
vocab_size=20000
tokenizer=Tokenizer(num_words=vocab_size)
tokenizer.fit_on_texts(train_list)
```

```
sequence1=tokenizer.texts_to_sequences(train_list1)
sequence2=tokenizer.texts_to_sequences(train_list2)
```

```
#padding the sequences to a constant size
max_length=100
sequence1=pad_sequences(sequence1,maxlen=max_length,padding='post')
sequence2=pad_sequences(sequence2,maxlen=max_length,padding='post')
```

```
train['seq1']=list(sequence1)
```

```
train['seq2']=list(sequence2)
```

```
train.head()
```

	id	qid1	qid2	question1	question2	is_duplicate	seq1	seq2
0	0	1	2	What is the step by step guide to invest in sh...	What is the step by step guide to invest in sh...	0	[2, 3, 1, 1222, 57, 1222, 2581, 7, 576, 8, 763...	[2, 3, 1, 1222, 57, 1222, 2581, 7, 576, 8, 763...
1	1	3	4	What is the story of Kohinoor (Koh-i-Noor) Dia...	What would happen if the Indian government sto...	0	[2, 3, 1, 559, 10, 14300, 13598, 5, 4565, 0, 0...	[2, 43, 182, 25, 1, 82, 237, 11296, 1, 14300, ...
2	2	5	6	How can I increase the speed of my internet co...	How can Internet speed be increased by hacking...	0	[4, 13, 5, 217, 1, 440, 10, 17, 361, 1827, 200...	[4, 13, 361, 440, 24, 3338, 57, 1344, 219, 109...
3	3	7	8	Why am I mentally very lonely? How can I solve...	Find the remainder when [math]23^{24}[/math] i...	0	[16, 72, 5, 2774, 312, 2757, 4, 13, 5, 649, 19...	[87, 1, 4170, 37, 230, 2234, 1343, 230, 3, 245...
4	4	9	10	Which one dissolve in water quikly sugar, salt...	Which fish would survive in salt water?	0	[23, 49, 7131, 8, 231, 1891, 2047, 10570, 12, ...	[23, 1945, 43, 1242, 8, 2047, 231, 0, 0, 0, 0,...

```
labels=np.asarray(train['is_duplicate'])
```

```
#functional API
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense,Embedding,LSTM,concatenate
```

Embedding the LSTM Layer

```
from tensorflow.keras import Input

text_input1=Input(shape=(None,),dtype='int32')
embedding1=Embedding(vocab_size,64)(text_input1)
encoded_text1=LSTM(32)(embedding1)

text_input2=Input(shape=(None,),dtype='int32')
embedding2=Embedding(vocab_size,64)(text_input2)
encoded_text2=LSTM(32)(embedding2)

concatenated=concatenate([encoded_text1,encoded_text2],axis=-1)

output=Dense(64,activation='relu')(concatenated)
output=Dense(1,activation='sigmoid')(output)
```

Compiling the model with Adam optimizer and loss as categorical crossentropy and evaluating the results using accuracy.

```
model=Model([text_input1,text_input2],output)
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])
```

```
model.summary()
```

```
Model: "model"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            [(None, None)]       0
__________________________________________________________________________________________________
input_2 (InputLayer)            [(None, None)]       0
__________________________________________________________________________________________________
embedding (Embedding)           (None, None, 64)     1280000     input_1[0][0]
__________________________________________________________________________________________________
embedding_1 (Embedding)         (None, None, 64)     1280000     input_2[0][0]
__________________________________________________________________________________________________
lstm (LSTM)                     (None, 32)           12416       embedding[0][0]
__________________________________________________________________________________________________
lstm_1 (LSTM)                   (None, 32)           12416       embedding_1[0][0]
__________________________________________________________________________________________________
concatenate (Concatenate)       (None, 64)           0           lstm[0][0]
                                                                 lstm_1[0][0]
__________________________________________________________________________________________________
dense (Dense)                   (None, 64)           4160        concatenate[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 1)            65          dense[0][0]
==================================================================================================
Total params: 2,589,057
Trainable params: 2,589,057
Non-trainable params: 0
__________________________________________________________________________________________________
```

Fitting the model

```
hist = model.fit([sequence1,sequence2],labels,epochs = 10,batch_size=128)
```

```
Epoch 1/10
3159/3159 [==============================] - 101s 32ms/step - loss: 0.6589 - accuracy: 0.6306
Epoch 2/10
3159/3159 [==============================] - 101s 32ms/step - loss: 0.6587 - accuracy: 0.6308
Epoch 3/10
3159/3159 [==============================] - 101s 32ms/step - loss: 0.6586 - accuracy: 0.6308
Epoch 4/10
3159/3159 [==============================] - 101s 32ms/step - loss: 0.6586 - accuracy: 0.6308
Epoch 5/10
3159/3159 [==============================] - 100s 32ms/step - loss: 0.6586 - accuracy: 0.6308
Epoch 6/10
3159/3159 [==============================] - 100s 32ms/step - loss: 0.6586 - accuracy: 0.6308
Epoch 7/10
3159/3159 [==============================] - 100s 32ms/step - loss: 0.6586 - accuracy: 0.6308
Epoch 8/10
3159/3159 [==============================] - 99s 31ms/step - loss: 0.6317 - accuracy: 0.6527
Epoch 9/10
3159/3159 [==============================] - 100s 32ms/step - loss: 0.5991 - accuracy: 0.6789
Epoch 10/10
3159/3159 [==============================] - 100s 32ms/step - loss: 0.5212 - accuracy: 0.7416
```

At the end of 10th epoch we are getting an accuracy of 74% and 0.52 loss.