```
# VAISHNAVI SOLANKAR
# MY PROGRAM

from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences


lines = ['It was a nice rainy day.', 'The things are so beautiful in his point.'
         'when your focus is clear, you won.', 'Many may happy returns of the da


tokenizer = Tokenizer()
tokenizer.fit_on_texts(lines)


tokenizer.word_docs
```

```
defaultdict(int,
            {'was': 1,
             'rainy': 1,
             'a': 1,
             'nice': 1,
             'day': 2,
             'it': 1,
             'beautiful': 1,
             'things': 1,
             'the': 2,
             'his': 1,
             'point': 1,
             'so': 1,
             'are': 1,
             'in': 1,
             'won': 1,
             'focus': 1,
             'clear': 1,
             'your': 1,
             'you': 1,
             'is': 1,
             'when': 1,
             'many': 1,
             'may': 1,
             'returns': 1,
             'happy': 1,
             'of': 1})
```

```
tokenizer.word_index
```

```
{'day': 1,
 'the': 2,
 'it': 3,
 'was': 4,
 'a': 5,
 'nice': 6,
 'rainy': 7,
 'things': 8,
 'are': 9,
 'so': 10,
 'beautiful': 11,
 'in': 12,
 'his': 13,
 'point': 14,
 'when': 15,
 'your': 16,
 'focus': 17,
 'is': 18,
 'clear': 19,
 'you': 20,
 'won': 21,
 'many': 22,
 'may': 23,
 'happy': 24,
 'returns': 25,
 'of': 26}
```

```
mat = tokenizer.texts_to_matrix(lines)
mat
```

```
array([[0., 1., 0., 1., 1., 1., 1., 1., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 1., 0., 0., 0., 0., 0., 1., 1., 1., 1., 1., 1., 1., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1.,
        1., 1., 1., 1., 1., 1., 0., 0., 0., 0., 0.],
       [0., 1., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 1., 1., 1., 1., 1.]])
```

```
seq = tokenizer.texts_to_sequences(lines)
seq
```

```
[[3, 4, 5, 6, 7, 1],
 [2, 8, 9, 10, 11, 12, 13, 14],
 [15, 16, 17, 18, 19, 20, 21],
 [22, 23, 24, 25, 26, 2, 1]]
```

```
padded = pad_sequences(seq, maxlen=10, padding='pre' )
padded
```

```
array([[ 0,  0,  0,  0,  3,  4,  5,  6,  7,  1],
       [ 0,  0,  2,  8,  9, 10, 11, 12, 13, 14],
       [ 0,  0,  0, 15, 16, 17, 18, 19, 20, 21],
       [ 0,  0,  0, 22, 23, 24, 25, 26,  2,  1]], dtype=int32)
```

Start coding or generate with AI.

```
import pandas as pd
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Embedding, LSTM
from sklearn.model_selection import train_test_split
import re
import numpy as np
import numpy as np
```

```
data = pd.read_csv('/content/twitter_sentiments.csv', names = ['id', 'loc', 'la
```

```
Start coding or generate with AI.
```

```
data.shape
```

```
(15837, 4)
```

```
data
```

	id	loc	label	text
0	2401	Borderlands	Positive	im getting on borderlands and i will murder yo...
1	2401	Borderlands	Positive	I am coming to the borders and I will kill you...
2	2401	Borderlands	Positive	im getting on borderlands and i will kill you ...
3	2401	Borderlands	Positive	im coming on borderlands and i will murder you...
4	2401	Borderlands	Positive	im getting on borderlands 2 and i will murder ...
...	...	...	...	...
15832	3108	Dota2	Negative	She wtf
15833	3108	Dota2	Negative	wtf
15834	3109	Dota2	Irrelevant	ICYMI: @BLASTDota finds sponsor in @KITKAT. . ...

```
data.dtypes
```

	0
id	int64
loc	object
label	object
text	object

dtype: object

```
data['text'] = data['text'].astype(str)
```

```
data.dtypes
data.dtypes
```

	0
id	int64
loc	object
label	object
text	object

dtype: object

```
def clean_text(text):
  text = text.lower()
  text = re.sub(r'[^a-zA-Z]+', ' ', text)
  return text
```

```
data["text"] = data["text"].apply(clean_text)
```

```
data
```

	id	loc	label	text
0	2401	Borderlands	Positive	im getting on borderlands and i will murder yo...
1	2401	Borderlands	Positive	i am coming to the borders and i will kill you...
2	2401	Borderlands	Positive	im getting on borderlands and i will kill you ...
3	2401	Borderlands	Positive	im coming on borderlands and i will murder you...
4	2401	Borderlands	Positive	im getting on borderlands and i will murder yo...
...	...	...	...	...
15832	3108	Dota2	Negative	she wtf
15833	3108	Dota2	Negative	wtf
15834	3109	Dota2	Irrelevant	icymi blastdota finds sponsor in kitkat we loo...

```
#Feature and target preparation
commments = data['text'].tolist()
target = data['label'].values
```

```
np.unique(target)
```

```
array(['Irrelevant', 'Negative', 'Neutral', 'Positive'], dtype=object)
```

```
#Tokenization and padding
Tokenizer = Tokenizer(num_words=5000)
Tokenizer.fit_on_texts(commments)
sequences =  Tokenizer.texts_to_sequences(commments)
padded_sequences = pad_sequences(sequences, maxlen=200)
```

```
padded_sequences
```

```
    array([[   0,    0,    0, ..., 1371,   13,   26],
           [   0,    0,    0, ...,  864,   13,   26],
           [   0,    0,    0, ...,  864,   13,   26],
           ...,
           [   0,    0,    0, ...,   73,  162,  125],
           [   0,    0,    0, ...,   73,  162,  125],
           [   0,    0,    0, ..., 1215,  162,  125]], dtype=int32)
```

```
padded_sequences.shape
```

```
    (15837, 200)
```

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(target)

from keras.utils import to_categorical
y_new = to_categorical(y)
```

```
y_new
```

```
    array([[0., 0., 0., 1.],
           [0., 0., 0., 1.],
           [0., 0., 0., 1.],
           ...,
           [1., 0., 0., 0.],
           [1., 0., 0., 0.],
           [1., 0., 0., 0.]])
```

```
y_new.shape
```

```
    (15837, 4)
```

```
from collections import Counter
Counter(target)
```

```
    Counter({'Positive': 4608,
             'Neutral': 3612,
             'Negative': 4674,
             'Irrelevant': 2943})
```

```
#Train_test_split / cross validation
x_train, x_test, y_train, y_test = train_test_split(padded_sequences, y_new, te
```

```
x_train.shape
```

```
    (12669, 200)
```

```
x_test.shape
```

```
    (3168, 200)
```

```
#Model definition
model = Sequential()
model.add(Embedding(5000, 100, input_length=200))
model.add(LSTM(64))
model.add(Dense(4, activation = 'softmax'))
#Multi label output with sigmoid activation
```

```
    /usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:
      warnings.warn(
```

```
#compile the model
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accu
```

```
#train the model
model.fit(x_train, y_train, validation_data=(x_test, y_test), epochs=5, batch_s
```

```
    Epoch 1/5
    396/396 ━━━━━━━━━━━━━━━━━━━━ 9s 11ms/step - accuracy: 0.4578 - loss: 1.1789
    Epoch 2/5
    396/396 ━━━━━━━━━━━━━━━━━━━━ 7s 11ms/step - accuracy: 0.8495 - loss: 0.3994
    Epoch 3/5
    396/396 ━━━━━━━━━━━━━━━━━━━━ 5s 11ms/step - accuracy: 0.9147 - loss: 0.2265
    Epoch 4/5
    396/396 ━━━━━━━━━━━━━━━━━━━━ 4s 10ms/step - accuracy: 0.9286 - loss: 0.1783
    Epoch 5/5
    396/396 ━━━━━━━━━━━━━━━━━━━━ 6s 12ms/step - accuracy: 0.9460 - loss: 0.1345
    <keras.src.callbacks.history.History at 0x7dc534f204c0>
```

```
#Prediction on unseen comment
new_comment = "I hate him."
new_sequence = tokenizer.texts_to_sequences([clean_text(new_comment)])
new_padded_sequence = pad_sequences(new_sequence, maxlen=200)
prediction = model.predict(new_padded_sequence)[0]
le.inverse_transform([np.argmax(prediction)])
```

```
    1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
    array(['Irrelevant'], dtype='<U10')
```

Start coding or generate with AI.