In [14]:

```
import os
import numpy as np
import tensorflow as tf
from tensorflow.keras.callbacks import ModelCheckpoint
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.layers import Embedding, LSTM, Dense, Bidirectional
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.models import Sequential
from tensorflow.keras.optimizers import Adam
```

In [4]:

```
file= open("data1.txt","r",encoding="UTF-8")
```

In [5]:

```
lines = []
for i in file:
      lines.append(i)
data =""
for i in lines:
      data =' '.join(lines)

data = data.replace('\n', '').replace('\r', '').replace('\ufeff', '').replace('“','').repla
print(len(data))
```

```
54373
```

In [6]:

```
data = data.split()
data = ' '.join(data)
print(len(data))
```

```
53458
```

In [28]:

```
data[:100]
```

Out[28]:

```
'The Project Gutenberg eBook of The Man With the Golden Eyes, by Alexander B
lade This eBook is for th'
```

In [30]:

```
import nltk
from nltk.tokenize import word_tokenize

tokens = word_tokenize(data)
train_len = 3+1
text_sequences = []
for i in range(train_len,len(tokens)):
    seq = tokens[i-train_len:i]
    text_sequences.append(seq)
print(text_sequences)
print(len(text_sequences))
```

```
[['The', 'Project', 'Gutenberg', 'eBook'], ['Project', 'Gutenberg', 'eBoo
k', 'of'], ['Gutenberg', 'eBook', 'of', 'The'], ['eBook', 'of', 'The', 'Ma
n'], ['of', 'The', 'Man', 'With'], ['The', 'Man', 'With', 'the'], ['Man',
'With', 'the', 'Golden'], ['With', 'the', 'Golden', 'Eyes'], ['the', 'Gold
en', 'Eyes', ','], ['Golden', 'Eyes', ',', 'by'], ['Eyes', ',', 'by', 'Ale
xander'], [',', 'by', 'Alexander', 'Blade'], ['by', 'Alexander', 'Blade',
'This'], ['Alexander', 'Blade', 'This', 'eBook'], ['Blade', 'This', 'eBoo
k', 'is'], ['This', 'eBook', 'is', 'for'], ['eBook', 'is', 'for', 'the'],
['is', 'for', 'the', 'use'], ['for', 'the', 'use', 'of'], ['the', 'use',
'of', 'anyone'], ['use', 'of', 'anyone', 'anywhere'], ['of', 'anyone', 'an
ywhere', 'in'], ['anyone', 'anywhere', 'in', 'the'], ['anywhere', 'in', 't
he', 'United'], ['in', 'the', 'United', 'States'], ['the', 'United', 'Stat
es', 'and'], ['United', 'States', 'and', 'most'], ['States', 'and', 'mos
t', 'other'], ['and', 'most', 'other', 'parts'], ['most', 'other', 'part
s', 'of'], ['other', 'parts', 'of', 'the'], ['parts', 'of', 'the', 'worl
d'], ['of', 'the', 'world', 'at'], ['the', 'world', 'at', 'no'], ['world',
'at', 'no', 'cost'], ['at', 'no', 'cost', 'and'], ['no', 'cost', 'and', 'w
ith'], ['cost', 'and', 'with', 'almost'], ['and', 'with', 'almost', 'no'],
['with', 'almost', 'no', 'restrictions'], ['almost', 'no', 'restrictions',
```

In [31]:

```
sequences = {}
count = 1
for i in range(len(tokens)):
    if tokens[i] not in sequences:
        sequences[tokens[i]] = count
        count += 1
print(sequences)
```

```
{'The': 1, 'Project': 2, 'Gutenberg': 3, 'eBook': 4, 'of': 5, 'Man': 6, 'W
ith': 7, 'the': 8, 'Golden': 9, 'Eyes': 10, ',': 11, 'by': 12, 'Alexande
r': 13, 'Blade': 14, 'This': 15, 'is': 16, 'for': 17, 'use': 18, 'anyone':
19, 'anywhere': 20, 'in': 21, 'United': 22, 'States': 23, 'and': 24, 'mos
t': 25, 'other': 26, 'parts': 27, 'world': 28, 'at': 29, 'no': 30, 'cost':
31, 'with': 32, 'almost': 33, 'restrictions': 34, 'whatsoever': 35, '.': 3
6, 'You': 37, 'may': 38, 'copy': 39, 'it': 40, 'give': 41, 'away': 42, 'o
r': 43, 're-use': 44, 'under': 45, 'terms': 46, 'License': 47, 'included':
48, 'this': 49, 'online': 50, 'www.gutenberg.org': 51, 'If': 52, 'you': 5
3, 'are': 54, 'not': 55, 'located': 56, 'will': 57, 'have': 58, 'to': 59,
'check': 60, 'laws': 61, 'country': 62, 'where': 63, 'before': 64, 'usin
g': 65, 'Title': 66, ':': 67, 'Author': 68, 'Release': 69, 'Date': 70, 'No
vember': 71, '23': 72, '2021': 73, '[': 74, '#': 75, '66802': 76, ']': 77,
'Language': 78, 'English': 79, 'Produced': 80, 'Greg': 81, 'Weeks': 82, 'M
ary': 83, 'Meehan': 84, 'Online': 85, 'Distributed': 86, 'Proofreading': 8
7, 'Team': 88, 'http': 89, '//www.pgdp.net': 90, '*': 91, 'START': 92, 'O
F': 93, 'THE': 94, 'PROJECT': 95, 'GUTENBERG': 96, 'EBOOK': 97, 'MAN': 98,
'WITH': 99, 'GOLDEN': 100, 'EYES': 101, 'Lee': 102, 'Hayden': 103, 'had':
104, 'sent': 105, 'eleven': 106, 'men': 107, 'their': 108, 'death': 109,
```

In [35]:

```
tokenizer = Tokenizer()
tokenizer.fit_on_texts(text_sequences)
sequence_data = tokenizer.texts_to_sequences(text_sequences)
print(sequence_data)
```

```
[[2, 20, 55, 113], [20, 55, 113, 7], [55, 113, 7, 2], [113, 7, 2, 28], [7,
2, 28, 16], [2, 28, 16, 2], [28, 16, 2, 79], [16, 2, 79, 51], [2, 79, 51,
3], [79, 51, 3, 43], [51, 3, 43, 426], [3, 43, 426, 427], [43, 426, 427, 2
3], [426, 427, 23, 113], [427, 23, 113, 25], [23, 113, 25, 27], [113, 25,
27, 2], [25, 27, 2, 123], [27, 2, 123, 7], [2, 123, 7, 169], [123, 7, 169,
428], [7, 169, 428, 11], [169, 428, 11, 2], [428, 11, 2, 101], [11, 2, 10
1, 80], [2, 101, 80, 8], [101, 80, 8, 225], [80, 8, 225, 66], [8, 225, 66,
574], [225, 66, 574, 7], [66, 574, 7, 2], [574, 7, 2, 226], [7, 2, 226, 4
9], [2, 226, 49, 45], [226, 49, 45, 331], [49, 45, 331, 8], [45, 331, 8, 1
6], [331, 8, 16, 277], [8, 16, 277, 45], [16, 277, 45, 575], [277, 45, 57
5, 576], [45, 575, 576, 1], [575, 576, 1, 10], [576, 1, 10, 68], [1, 10, 6
8, 124], [10, 68, 124, 24], [68, 124, 24, 3], [124, 24, 3, 227], [24, 3, 2
27, 24], [3, 227, 24, 170], [227, 24, 170, 21], [24, 170, 21, 577], [170,
21, 577, 24], [21, 577, 24, 171], [577, 24, 171, 2], [24, 171, 2, 69], [17
1, 2, 69, 7], [2, 69, 7, 2], [69, 7, 2, 20], [7, 2, 20, 55], [2, 20, 55, 9
2], [20, 55, 92, 278], [55, 92, 278, 16], [92, 278, 16, 23], [278, 16, 23,
113], [16, 23, 113, 21], [23, 113, 21, 279], [113, 21, 279, 49], [21, 279,
49, 280], [279, 49, 280, 1], [49, 280, 1, 53], [280, 1, 53, 10], [1, 53, 1
0, 33], [53, 10, 33, 26], [10, 33, 26, 198], [33, 26, 198, 11], [26, 198,
```

In [36]:

```
len(sequence_data)
```

Out[36]:

```
11458
```

In [37]:

```
sequence_data[:10]
```

Out[37]:

```
[[2, 20, 55, 113],
 [20, 55, 113, 7],
 [55, 113, 7, 2],
 [113, 7, 2, 28],
 [7, 2, 28, 16],
 [2, 28, 16, 2],
 [28, 16, 2, 79],
 [16, 2, 79, 51],
 [2, 79, 51, 3],
 [79, 51, 3, 43]]
```

In [38]:

```
total_words = len(tokenizer.word_index) + 1
print(total_words)
```

```
2017
```

In [11]:

```
input_sequences = []
for line in lines:
    token_list = tokenizer.texts_to_sequences([line])[0]
    #print(token_list)

    for i in range(1, len(token_list)):
        n_gram_sequence = token_list[:i+1]
        input_sequences.append(n_gram_sequence)

# print(input_sequences)
print("Total input sequences: ", len(input_sequences))
```

```
Total input sequences:  8540
```

In [12]:

```
input_sequences[:15]
```

Out[12]:

```
[[1, 14],
 [1, 14, 11],
 [1, 14, 11, 103],
 [1, 14, 11, 103, 3],
 [1, 14, 11, 103, 3, 1],
 [1, 14, 11, 103, 3, 1, 23],
 [1, 14, 11, 103, 3, 1, 23, 12],
 [1, 14, 11, 103, 3, 1, 23, 12, 1],
 [1, 14, 11, 103, 3, 1, 23, 12, 1, 72],
 [1, 14, 11, 103, 3, 1, 23, 12, 1, 72, 44],
 [1, 14, 11, 103, 3, 1, 23, 12, 1, 72, 44, 37],
 [1, 14, 11, 103, 3, 1, 23, 12, 1, 72, 44, 37, 421],
 [17, 103],
 [17, 103, 19],
 [17, 103, 19, 21]]
```

In [15]:

```
max_sequence_len = max([len(x) for x in input_sequences])
input_sequences = np.array(pad_sequences(input_sequences, maxlen=max_sequence_len, padding=
input_sequences[1]
```

Out[15]:

```
array([ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  1, 14, 11])
```

In [16]:

```
xs, labels = input_sequences[:,:-1],input_sequences[:,-1]
ys = tf.keras.utils.to_categorical(labels, num_classes=total_words)
```

In [17]:

```
print(xs[5])
print(labels[5])
print(ys[5][14])
```

```
[  0   0   0   0   0   0   0   0   0   0   1  14  11 103   3   1]
23
0.0
```

Training part

In [23]:

```
model = Sequential()
model.add(Embedding(total_words, 100, input_length=max_sequence_len-1))
model.add(Bidirectional(LSTM(150)))
model.add(Dense(total_words, activation='softmax'))
adam = Adam(lr=0.01)
checkpoint = ModelCheckpoint("next_words.h5", monitor='loss', verbose=1, save_best_only=Tru
model.compile(loss='categorical_crossentropy', optimizer=adam, metrics=['accuracy'])
history = model.fit(xs, ys, epochs=50, verbose=1,callbacks=[checkpoint])
#print model.summary()
print(model)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\keras\optimizer_v2\adam.py:105: U
serWarning: The `lr` argument is deprecated, use `learning_rate` instead.
  super(Adam, self).__init__(name, **kwargs)

Epoch 1/50
267/267 [==============================] - ETA: 0s - loss: 6.2691 - accurac
y: 0.0831
Epoch 00001: loss improved from inf to 6.26914, saving model to next_words.h
5
267/267 [==============================] - 14s 36ms/step - loss: 6.2691 - ac
curacy: 0.0831
Epoch 2/50
266/267 [============================>.] - ETA: 0s - loss: 5.2280 - accurac
y: 0.1467
Epoch 00002: loss improved from 6.26914 to 5.22423, saving model to next_wor
ds.h5
267/267 [==============================] - 8s 32ms/step - loss: 5.2242 - acc
uracy: 0.1470
Epoch 3/50
267/267 [==============================] - ETA: 0s - loss: 4.2855 - accurac
y: 0.2049
Epoch 00003: loss improved from 5.22423 to 4.28548, saving model to next_wor
ds.h5
267/267 [==============================] - 8s 30ms/step - loss: 4.2855 - acc
uracy: 0.2049
Epoch 4/50
266/267 [============================>.] - ETA: 0s - loss: 3.3288 - accurac
y: 0.2946
Epoch 00004: loss improved from 4.28548 to 3.33028, saving model to next_wor
ds.h5
267/267 [==============================] - 8s 31ms/step - loss: 3.3303 - acc
uracy: 0.2946
Epoch 5/50
266/267 [============================>.] - ETA: 0s - loss: 2.4931 - accurac
y: 0.4166
Epoch 00005: loss improved from 3.33028 to 2.49628, saving model to next_wor
ds.h5
267/267 [==============================] - 8s 31ms/step - loss: 2.4963 - acc
uracy: 0.4157
Epoch 6/50
266/267 [============================>.] - ETA: 0s - loss: 1.8953 - accurac
y: 0.5338
Epoch 00006: loss improved from 2.49628 to 1.89732, saving model to next_wor
ds.h5
267/267 [==============================] - 8s 31ms/step - loss: 1.8973 - acc
uracy: 0.5337
Epoch 7/50
267/267 [==============================] - ETA: 0s - loss: 1.4541 - accurac
```

```
y: 0.6327
Epoch 00007: loss improved from 1.89732 to 1.45406, saving model to next_wor
ds.h5
267/267 [==============================] - 9s 33ms/step - loss: 1.4541 - acc
uracy: 0.6327
Epoch 8/50
267/267 [==============================] - ETA: 0s - loss: 1.0998 - accurac
y: 0.7242
Epoch 00008: loss improved from 1.45406 to 1.09976, saving model to next_wor
ds.h5
267/267 [==============================] - 9s 33ms/step - loss: 1.0998 - acc
uracy: 0.7242
Epoch 9/50
267/267 [==============================] - ETA: 0s - loss: 0.8713 - accurac
y: 0.7806
Epoch 00009: loss improved from 1.09976 to 0.87126, saving model to next_wor
ds.h5
267/267 [==============================] - 8s 31ms/step - loss: 0.8713 - acc
uracy: 0.7806
Epoch 10/50
267/267 [==============================] - ETA: 0s - loss: 0.7940 - accurac
y: 0.7970
Epoch 00010: loss improved from 0.87126 to 0.79397, saving model to next_wor
ds.h5
267/267 [==============================] - 8s 32ms/step - loss: 0.7940 - acc
uracy: 0.7970
Epoch 11/50
267/267 [==============================] - ETA: 0s - loss: 0.7400 - accurac
y: 0.8046
Epoch 00011: loss improved from 0.79397 to 0.74004, saving model to next_wor
ds.h5
267/267 [==============================] - 8s 31ms/step - loss: 0.7400 - acc
uracy: 0.8046
Epoch 12/50
267/267 [==============================] - ETA: 0s - loss: 0.6966 - accurac
y: 0.8183
Epoch 00012: loss improved from 0.74004 to 0.69657, saving model to next_wor
ds.h5
267/267 [==============================] - 8s 31ms/step - loss: 0.6966 - acc
uracy: 0.8183
Epoch 13/50
266/267 [============================>.] - ETA: 0s - loss: 0.6855 - accurac
y: 0.8213
Epoch 00013: loss improved from 0.69657 to 0.68564, saving model to next_wor
ds.h5
267/267 [==============================] - 8s 29ms/step - loss: 0.6856 - acc
uracy: 0.8213
Epoch 14/50
266/267 [============================>.] - ETA: 0s - loss: 0.7529 - accurac
y: 0.7976
Epoch 00014: loss did not improve from 0.68564
267/267 [==============================] - 8s 30ms/step - loss: 0.7530 - acc
uracy: 0.7978
Epoch 15/50
266/267 [============================>.] - ETA: 0s - loss: 0.8352 - accurac
y: 0.7768
Epoch 00015: loss did not improve from 0.68564
267/267 [==============================] - 9s 32ms/step - loss: 0.8355 - acc
uracy: 0.7763
Epoch 16/50
267/267 [==============================] - ETA: 0s - loss: 0.8329 - accurac
```

```
y: 0.7731
Epoch 00016: loss did not improve from 0.68564
267/267 [==============================] - 8s 31ms/step - loss: 0.8329 - acc
uracy: 0.7731
Epoch 17/50
267/267 [==============================] - ETA: 0s - loss: 0.8192 - accurac
y: 0.7714
Epoch 00017: loss did not improve from 0.68564
267/267 [==============================] - 8s 29ms/step - loss: 0.8192 - acc
uracy: 0.7714
Epoch 18/50
267/267 [==============================] - ETA: 0s - loss: 0.7506 - accurac
y: 0.7929
Epoch 00018: loss did not improve from 0.68564
267/267 [==============================] - 8s 28ms/step - loss: 0.7506 - acc
uracy: 0.7929
Epoch 19/50
267/267 [==============================] - ETA: 0s - loss: 0.6741 - accurac
y: 0.8142
Epoch 00019: loss improved from 0.68564 to 0.67409, saving model to next_wor
ds.h5
267/267 [==============================] - 8s 30ms/step - loss: 0.6741 - acc
uracy: 0.8142
Epoch 20/50
266/267 [============================>.] - ETA: 0s - loss: 0.5827 - accurac
y: 0.8410
Epoch 00020: loss improved from 0.67409 to 0.58290, saving model to next_wor
ds.h5
267/267 [==============================] - 10s 36ms/step - loss: 0.5829 - ac
curacy: 0.8410
Epoch 21/50
266/267 [============================>.] - ETA: 0s - loss: 0.5317 - accurac
y: 0.8536
Epoch 00021: loss improved from 0.58290 to 0.53110, saving model to next_wor
ds.h5
267/267 [==============================] - 9s 33ms/step - loss: 0.5311 - acc
uracy: 0.8537
Epoch 22/50
266/267 [============================>.] - ETA: 0s - loss: 0.5009 - accurac
y: 0.8617
Epoch 00022: loss improved from 0.53110 to 0.50350, saving model to next_wor
ds.h5
267/267 [==============================] - 8s 30ms/step - loss: 0.5035 - acc
uracy: 0.8610
Epoch 23/50
266/267 [============================>.] - ETA: 0s - loss: 0.5421 - accurac
y: 0.8479
Epoch 00023: loss did not improve from 0.50350
267/267 [==============================] - 8s 30ms/step - loss: 0.5436 - acc
uracy: 0.8474
Epoch 24/50
267/267 [==============================] - ETA: 0s - loss: 0.6298 - accurac
y: 0.8214
Epoch 00024: loss did not improve from 0.50350
267/267 [==============================] - 8s 29ms/step - loss: 0.6298 - acc
uracy: 0.8214
Epoch 25/50
267/267 [==============================] - ETA: 0s - loss: 0.7747 - accurac
y: 0.7810
Epoch 00025: loss did not improve from 0.50350
267/267 [==============================] - 8s 29ms/step - loss: 0.7747 - acc
```

```
uracy: 0.7810
Epoch 26/50
267/267 [==============================] - ETA: 0s - loss: 0.8767 - accurac
y: 0.7532
Epoch 00026: loss did not improve from 0.50350
267/267 [==============================] - 8s 30ms/step - loss: 0.8767 - acc
uracy: 0.7532
Epoch 27/50
267/267 [==============================] - ETA: 0s - loss: 0.9288 - accurac
y: 0.7352
Epoch 00027: loss did not improve from 0.50350
267/267 [==============================] - 8s 29ms/step - loss: 0.9288 - acc
uracy: 0.7352
Epoch 28/50
267/267 [==============================] - ETA: 0s - loss: 0.8078 - accurac
y: 0.7720
Epoch 00028: loss did not improve from 0.50350
267/267 [==============================] - 8s 28ms/step - loss: 0.8078 - acc
uracy: 0.7720
Epoch 29/50
267/267 [==============================] - ETA: 0s - loss: 0.6311 - accurac
y: 0.8224
Epoch 00029: loss did not improve from 0.50350
267/267 [==============================] - 8s 28ms/step - loss: 0.6311 - acc
uracy: 0.8224
Epoch 30/50
267/267 [==============================] - ETA: 0s - loss: 0.4963 - accurac
y: 0.8567
Epoch 00030: loss improved from 0.50350 to 0.49629, saving model to next_wor
ds.h5
267/267 [==============================] - 10s 38ms/step - loss: 0.4963 - ac
curacy: 0.8567
Epoch 31/50
267/267 [==============================] - ETA: 0s - loss: 0.4110 - accurac
y: 0.8838
Epoch 00031: loss improved from 0.49629 to 0.41104, saving model to next_wor
ds.h5
267/267 [==============================] - 9s 32ms/step - loss: 0.4110 - acc
uracy: 0.8838
Epoch 32/50
266/267 [============================>.] - ETA: 0s - loss: 0.3562 - accurac
y: 0.9007 ETA:
Epoch 00032: loss improved from 0.41104 to 0.35740, saving model to next_wor
ds.h5
267/267 [==============================] - 10s 36ms/step - loss: 0.3574 - ac
curacy: 0.9006
Epoch 33/50


266/267 [============================>.] - ETA: 0s - loss: 0.3097 - accura
cy: 0.9127
Epoch 00033: loss improved from 0.35740 to 0.30997, saving model to next_w
ords.h5
267/267 [==============================] - 9s 32ms/step - loss: 0.3100 - a
ccuracy: 0.9126
Epoch 34/50
266/267 [============================>.] - ETA: 0s - loss: 0.2915 - accura
cy: 0.9166
Epoch 00034: loss improved from 0.30997 to 0.29133, saving model to next_w
ords.h5
267/267 [==============================] - 9s 32ms/step - loss: 0.2913 - a
```

```
ccuracy: 0.9167
Epoch 35/50
266/267 [============================>.] - ETA: 0s - loss: 0.2784 - accura
cy: 0.9182
Epoch 00035: loss improved from 0.29133 to 0.27828, saving model to next_w
ords.h5
267/267 [==============================] - 9s 34ms/step - loss: 0.2783 - a
ccuracy: 0.9183
Epoch 36/50
267/267 [==============================] - ETA: 0s - loss: 0.2756 - accura
cy: 0.9183
Epoch 00036: loss improved from 0.27828 to 0.27562, saving model to next_w
ords.h5
267/267 [==============================] - 10s 36ms/step - loss: 0.2756 -
accuracy: 0.9183
Epoch 37/50
267/267 [==============================] - ETA: 0s - loss: 0.2670 - accura
cy: 0.9205
Epoch 00037: loss improved from 0.27562 to 0.26703, saving model to next_w
ords.h5
267/267 [==============================] - 9s 34ms/step - loss: 0.2670 - a
ccuracy: 0.9205
Epoch 38/50
266/267 [============================>.] - ETA: 0s - loss: 0.2689 - accura
cy: 0.9208
Epoch 00038: loss did not improve from 0.26703
267/267 [==============================] - 9s 32ms/step - loss: 0.2688 - a
ccuracy: 0.9208
Epoch 39/50
266/267 [============================>.] - ETA: 0s - loss: 0.2656 - accura
cy: 0.9202
Epoch 00039: loss improved from 0.26703 to 0.26606, saving model to next_w
ords.h5
267/267 [==============================] - 9s 34ms/step - loss: 0.2661 - a
ccuracy: 0.9201
Epoch 40/50
267/267 [==============================] - ETA: 0s - loss: 0.2664 - accura
cy: 0.9203
Epoch 00040: loss did not improve from 0.26606
267/267 [==============================] - 9s 34ms/step - loss: 0.2664 - a
ccuracy: 0.9203
Epoch 41/50
267/267 [==============================] - ETA: 0s - loss: 0.2701 - accura
cy: 0.9177
Epoch 00041: loss did not improve from 0.26606
267/267 [==============================] - 9s 34ms/step - loss: 0.2701 - a
ccuracy: 0.9177
Epoch 42/50
267/267 [==============================] - ETA: 0s - loss: 0.3641 - accura
cy: 0.8959
Epoch 00042: loss did not improve from 0.26606
267/267 [==============================] - 9s 33ms/step - loss: 0.3641 - a
ccuracy: 0.8959
Epoch 43/50
267/267 [==============================] - ETA: 0s - loss: 2.9882 - accura
cy: 0.4204
Epoch 00043: loss did not improve from 0.26606
267/267 [==============================] - 9s 33ms/step - loss: 2.9882 - a
ccuracy: 0.4204
Epoch 44/50
266/267 [============================>.] - ETA: 0s - loss: 2.4254 - accura
```

```
cy: 0.4800
Epoch 00044: loss did not improve from 0.26606
267/267 [==============================] - 9s 32ms/step - loss: 2.4280 - a
ccuracy: 0.4797
Epoch 45/50
266/267 [============================>.] - ETA: 0s - loss: 1.4504 - accura
cy: 0.6330
Epoch 00045: loss did not improve from 0.26606
267/267 [==============================] - 9s 32ms/step - loss: 1.4513 - a
ccuracy: 0.6327
Epoch 46/50
267/267 [==============================] - ETA: 0s - loss: 0.9469 - accura
cy: 0.7410 ETA: 0s - los
Epoch 00046: loss did not improve from 0.26606
267/267 [==============================] - 9s 33ms/step - loss: 0.9469 - a
ccuracy: 0.7410
Epoch 47/50
266/267 [============================>.] - ETA: 0s - loss: 0.6466 - accura
cy: 0.8179
Epoch 00047: loss did not improve from 0.26606
267/267 [==============================] - 9s 33ms/step - loss: 0.6456 - a
ccuracy: 0.8183
Epoch 48/50
267/267 [==============================] - ETA: 0s - loss: 0.4698 - accura
cy: 0.8718
Epoch 00048: loss did not improve from 0.26606
267/267 [==============================] - 9s 34ms/step - loss: 0.4698 - a
ccuracy: 0.8718
Epoch 49/50
267/267 [==============================] - ETA: 0s - loss: 0.3621 - accura
cy: 0.9026
Epoch 00049: loss did not improve from 0.26606
267/267 [==============================] - 9s 34ms/step - loss: 0.3621 - a
ccuracy: 0.9026
Epoch 50/50
266/267 [============================>.] - ETA: 0s - loss: 0.3168 - accura
cy: 0.9133
Epoch 00050: loss did not improve from 0.26606
267/267 [==============================] - 9s 32ms/step - loss: 0.3167 - a
ccuracy: 0.9133
<keras.engine.sequential.Sequential object at 0x0000029F709B6A60>
```

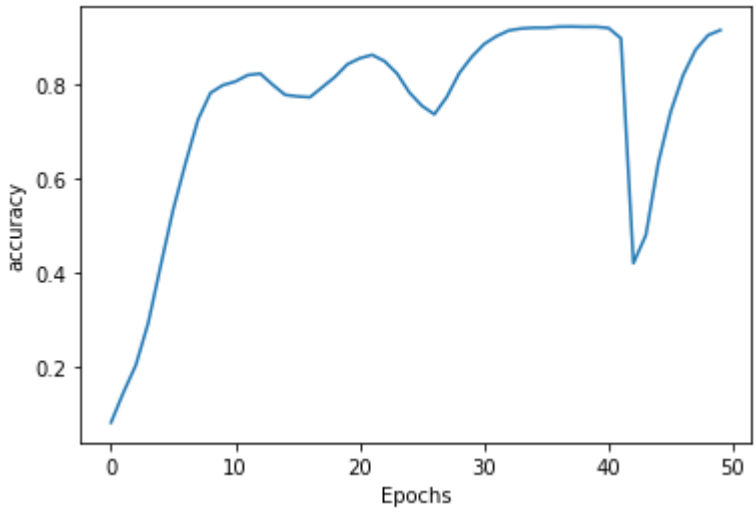
In [24]:

```
import matplotlib.pyplot as plt


def plot_graphs(history, string):
    plt.plot(history.history[string])
    plt.xlabel("Epochs")
    plt.ylabel(string)
    plt.show()
```

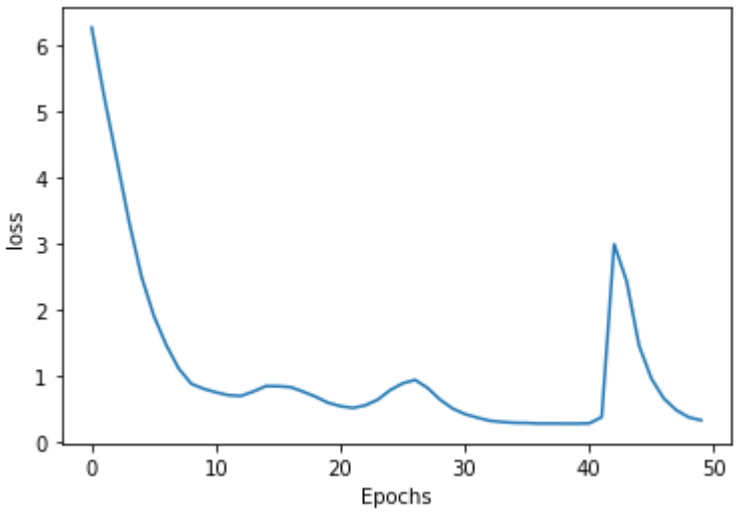
In [25]:

```
plot_graphs(history, 'accuracy')
```



In [26]:

```
plot_graphs(history, 'loss')
```



In [27]:

```
history.history.keys()
```

Out[27]:

```
dict_keys(['loss', 'accuracy'])
```

Predicting the output (Next two continous words)

In [18]:

```
from tensorflow.keras.models import load_model
import pickle

# Load the model and tokenizer
model = load_model('next_words.h5')
```

In [19]:

```
while(True):
    seed_text = input("Enter your line: ")

    if seed_text == "0":
        print("Execution completed.....")
        break

    else:
        try:
            next_words = 2
            suggested_word = []
            #temp = seed_text
            for _ in range(next_words):

                token_list = tokenizer.texts_to_sequences([seed_text])[0]
                #print(token_list)
                token_list = pad_sequences([token_list], maxlen=max_sequence_len-1, padding
                predicted = np.argmax(model.predict(token_list), axis=-1)
                output_word = ""

                for word, index in tokenizer.word_index.items():
                    if index == predicted:
                        output_word = word
                        suggested_word.append(output_word)
                        break

                seed_text += " " + output_word
            print("Suggested next two word are : ",suggested_word)
            #n = int(input())

            print(seed_text)
        except Exception as e:
            print("Error occurred: ",e)
        continue
```

```
Enter your line: the output of the project
Suggested next two word are :  ['gutenberg', 'literary']
the output of the project  gutenberg literary
Enter your line: the output of the project  gutenberg literary
Suggested next two word are :  ['archive', 'foundation']
the output of the project  gutenberg literary archive foundation
Enter your line: the output of the project  gutenberg literary archive fou
ndation
Suggested next two word are :  ['is', 'a']
the output of the project  gutenberg literary archive foundation is a
Enter your line: the output of the project  gutenberg literary archive fou
ndation is a
Suggested next two word are :  ['non', 'profit']
the output of the project  gutenberg literary archive foundation is a non
profit
Enter your line: the output of the project  gutenberg literary archive fou
ndation is a non profit
Suggested next two word are :  ['but', 'a']
the output of the project  gutenberg literary archive foundation is a non
profit but a
Enter your line: the output of the project  gutenberg literary archive fou
ndation is a non profit but a
Suggested next two word are :  ['non', 'profit']
```

```
the output of the project  gutenberg literary archive foundation is a non
profit but a non profit
Enter your line: 0
Execution completed.....
```

Predicting the three suggested words

In [20]:

```
from keras.preprocessing.sequence import pad_sequences
input_text = input().strip().lower()
encoded_text = tokenizer.texts_to_sequences([input_text])[0]
pad_encoded = pad_sequences([encoded_text], maxlen=max_sequence_len-1, truncating='pre')
print(encoded_text, pad_encoded)
for i in (model.predict(pad_encoded)[0]).argsort()[-3:][::-1]:
    pred_word = tokenizer.index_word[i]
    print("Next word suggestion:",pred_word)
```

```
next word can be
[213, 54, 43] [[  0   0   0   0   0   0   0   0   0   0   0   0   0 213  54
43]]
Next word suggestion: only
Next word suggestion: there
Next word suggestion: copied
```

In [21]:

```

```

In [ ]:

```

```

In [ ]:

```

```