Recurrent neural networks

Practice Session 11

Changho Suh

January 25, 2024

Outline

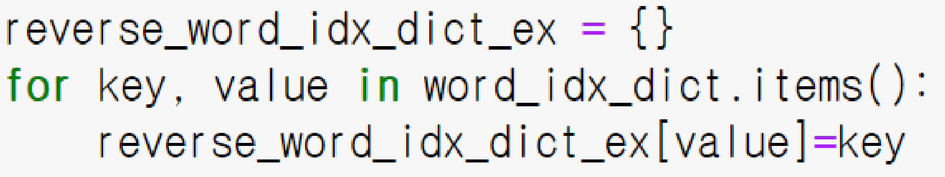
Will learn more about IMDB dataset:

- Conversion from numerical values to words

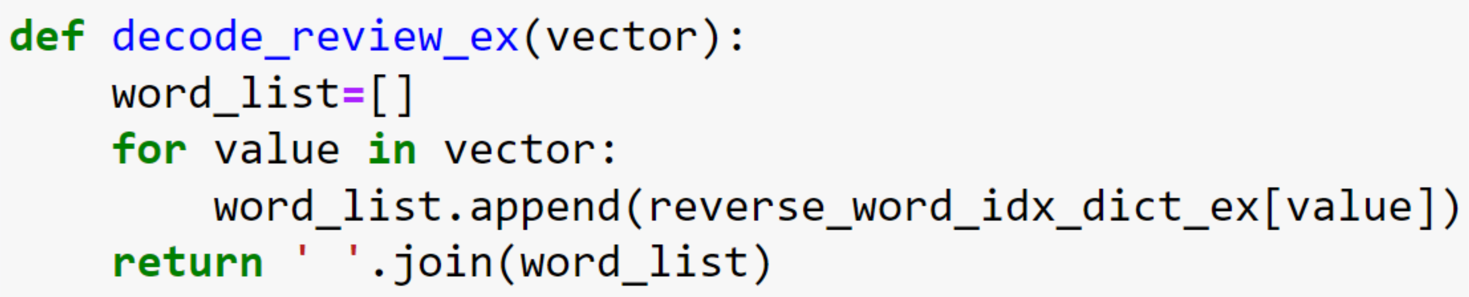
Will do data processing:

- Zero padding
- Word embedding

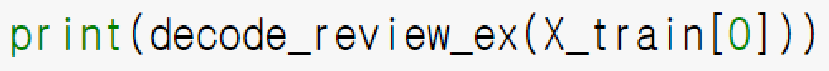
Conversion without considering “+3”



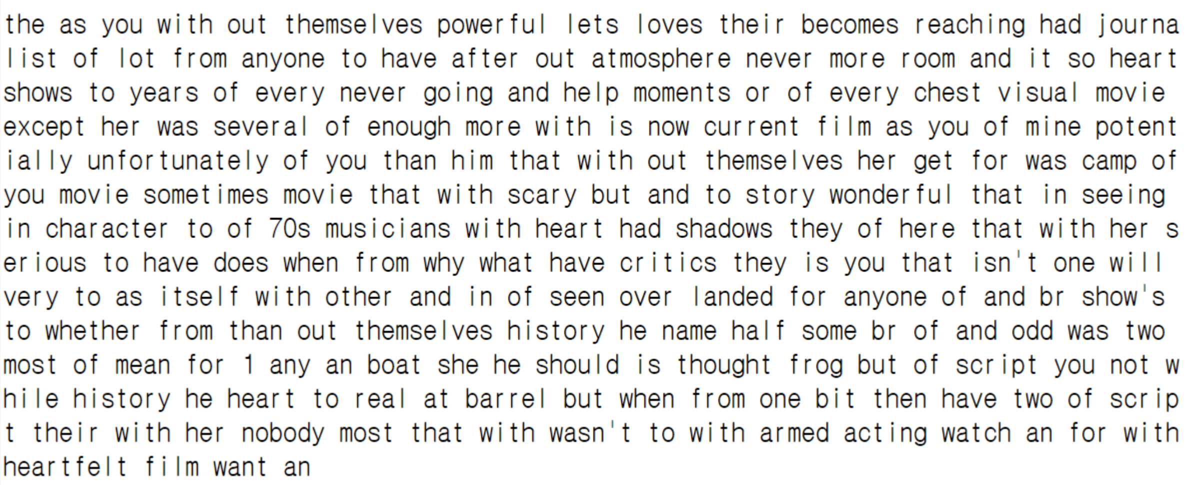
```
reverse_word_idx_dict_ex = {}
for key, value in word_idx_dict.items():
    reverse_word_idx_dict_ex[value]=key
```



```
def decode_review_ex(vector):
    word_list=[]
    for value in vector:
        word_list.append(reverse_word_idx_dict_ex[value])
    return ' '.join(word_list)
```

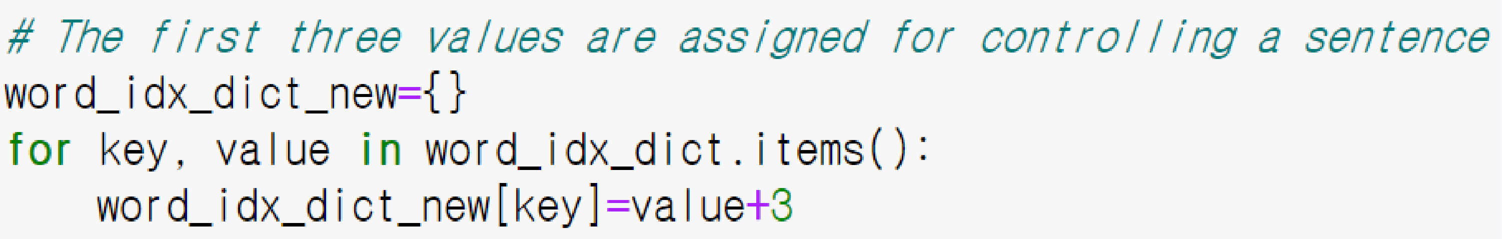


```
print(decode_review_ex(X_train[0]))
```

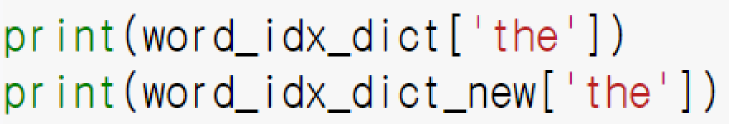


```
the as you with out themselves powerful lets loves their becomes reaching had journa
list of lot from anyone to have after out atmosphere never more room and it so heart
shows to years of every never going and help moments or of every chest visual movie
except her was several of enough more with is now current film as you of mine potent
ially unfortunately of you than him that with out themselves her get for was camp of
you movie sometimes movie that with scary but and to story wonderful that in seeing
in character to of 70s musicians with heart had shadows they of here that with her s
erious to have does when from why what have critics they is you that isn't one will
very to as itself with other and in of seen over landed for anyone of and br show's
to whether from than out themselves history he name half some br of and odd was two
most of mean for 1 any an boat she he should is thought frog but of script you not w
hile history he heart to real at barrel but when from one bit then have two of scrip
t their with her nobody most that with wasn't to with armed acting watch an for with
heartfelt film want an
```

Conversion with considering “+3”

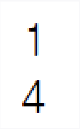


```
# The first three values are assigned for controlling a sentence
word_idx_dict_new={}
for key, value in word_idx_dict.items():
    word_idx_dict_new[key]=value+3
```

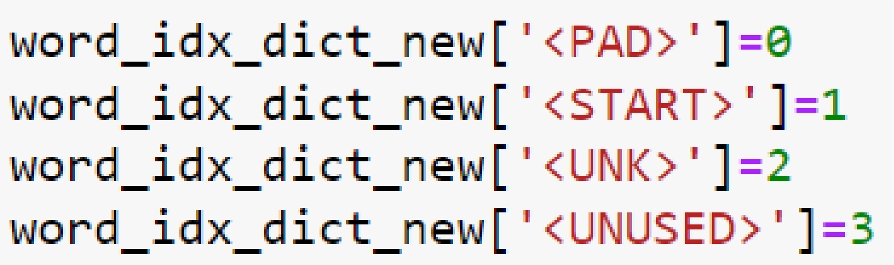


```
print(word_idx_dict['the'])
print(word_idx_dict_new['the'])
```

→

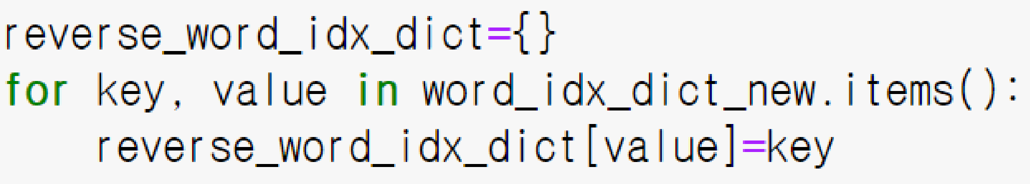


```
1
4
```

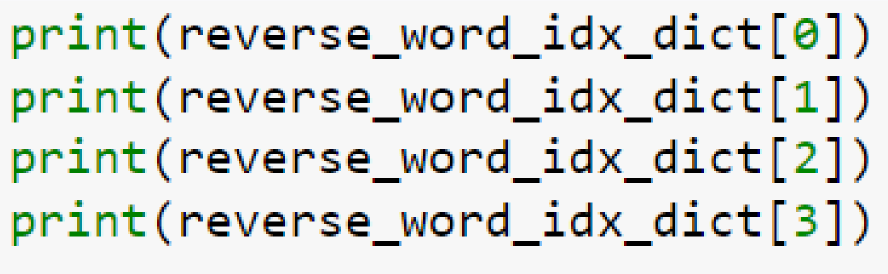


```
word_idx_dict_new['<PAD>']=0
word_idx_dict_new['<START>']=1
word_idx_dict_new['<UNK>']=2
word_idx_dict_new['<UNUSED>']=3
```

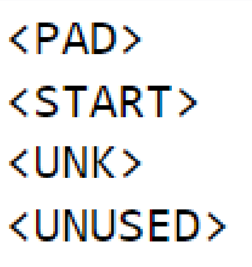
Conversion with considering “+3”



```python
reverse_word_idx_dict={}
for key, value in word_idx_dict_new.items():
    reverse_word_idx_dict[value]=key
```

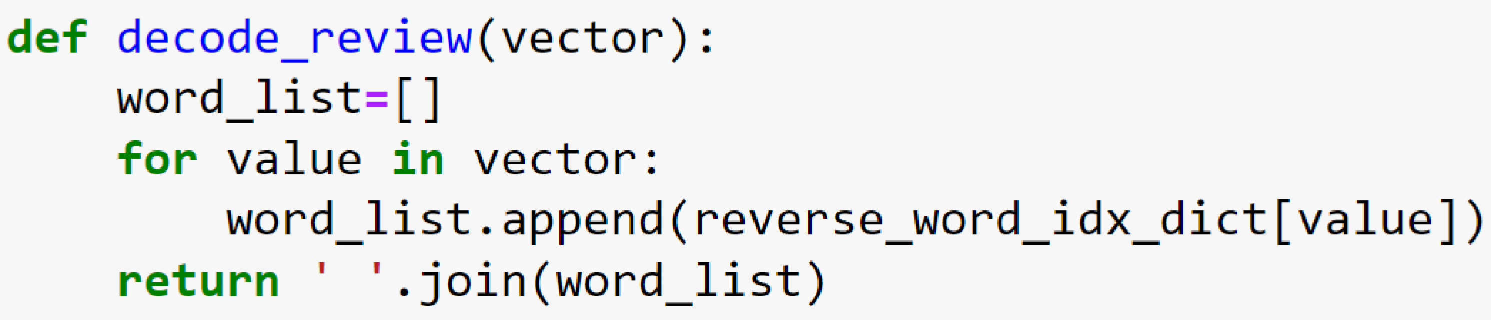


```python
print(reverse_word_idx_dict[0])
print(reverse_word_idx_dict[1])
print(reverse_word_idx_dict[2])
print(reverse_word_idx_dict[3])
```

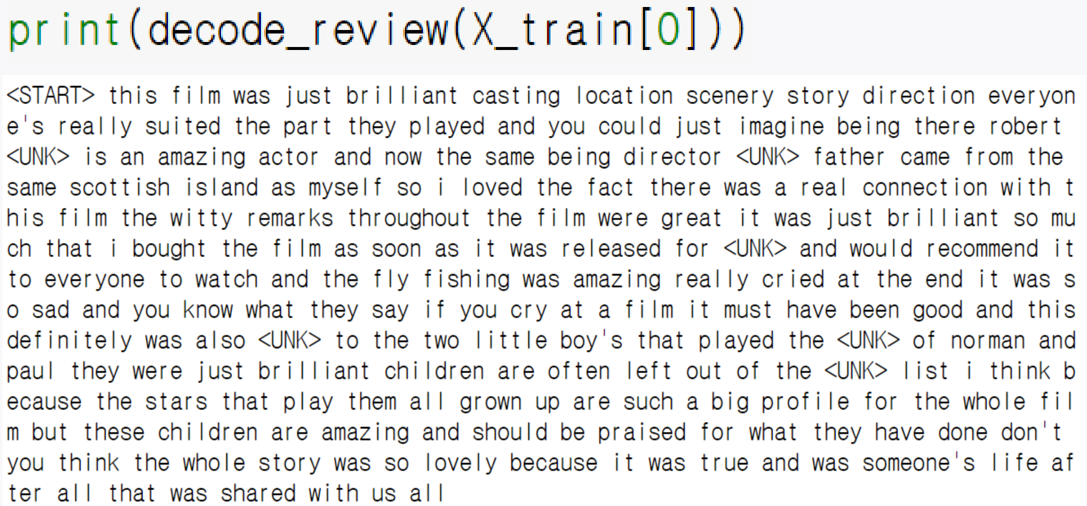


```
<PAD>
<START>
<UNK>
<UNUSED>
```

Conversion with considering “+3”

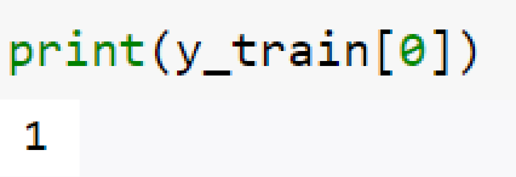


```python
def decode_review(vector):
    word_list=[]
    for value in vector:
        word_list.append(reverse_word_idx_dict[value])
    return ' '.join(word_list)
```



```python
print(decode_review(X_train[0]))
```

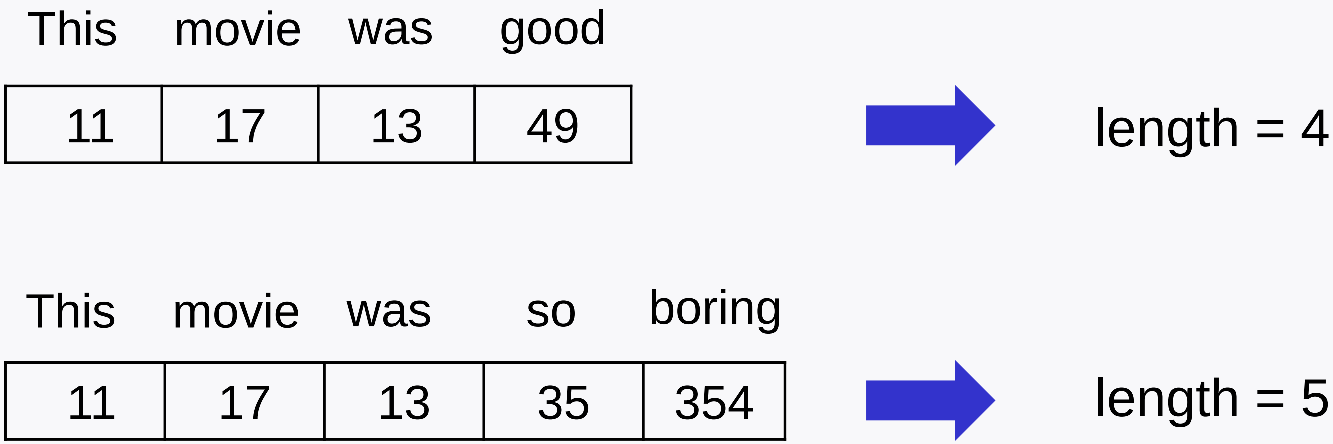
```
<START> this film was just brilliant casting location scenery story direction everyon
e's really suited the part they played and you could just imagine being there robert
<UNK> is an amazing actor and now the same being director <UNK> father came from the
same scottish island as myself so i loved the fact there was a real connection with t
his film the witty remarks throughout the film were great it was just brilliant so mu
ch that i bought the film as soon as it was released for <UNK> and would recommend it
to everyone to watch and the fly fishing was amazing really cried at the end it was s
o sad and you know what they say if you cry at a film it must have been good and this
definitely was also <UNK> to the two little boy's that played the <UNK> of norman and
paul they were just brilliant children are often left out of the <UNK> list i think b
ecause the stars that play them all grown up are such a big profile for the whole fil
m but these children are amazing and should be praised for what they have done don't
you think the whole story was so lovely because it was true and was someone's life af
ter all that was shared with us all
```



```python
print(y_train[0])
```

```
1
```

Data preprocessing



This	movie	was	good
11	17	13	49

length = 4

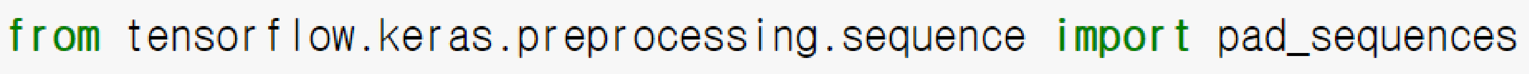
This	movie	was	so	boring
11	17	13	35	354

length = 5

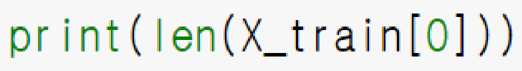
How to deal with different lengths?

→ **Padding**

Padding

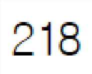


```
from tensorflow.keras.preprocessing.sequence import pad_sequences
```

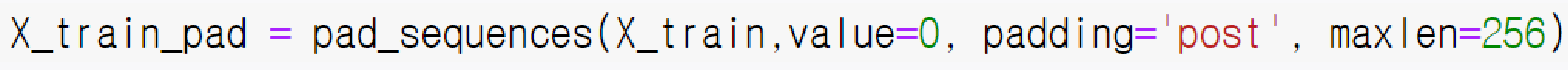


```
print(len(X_train[0]))
```

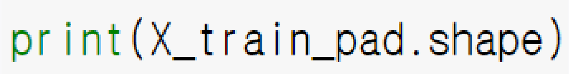
→



```
218
```

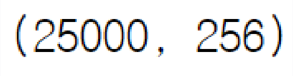


```
X_train_pad = pad_sequences(X_train,value=0, padding='post', maxlen=256)
```

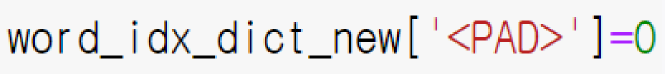


```
print(X_train_pad.shape)
```

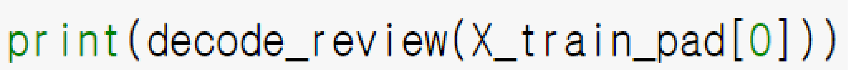
→



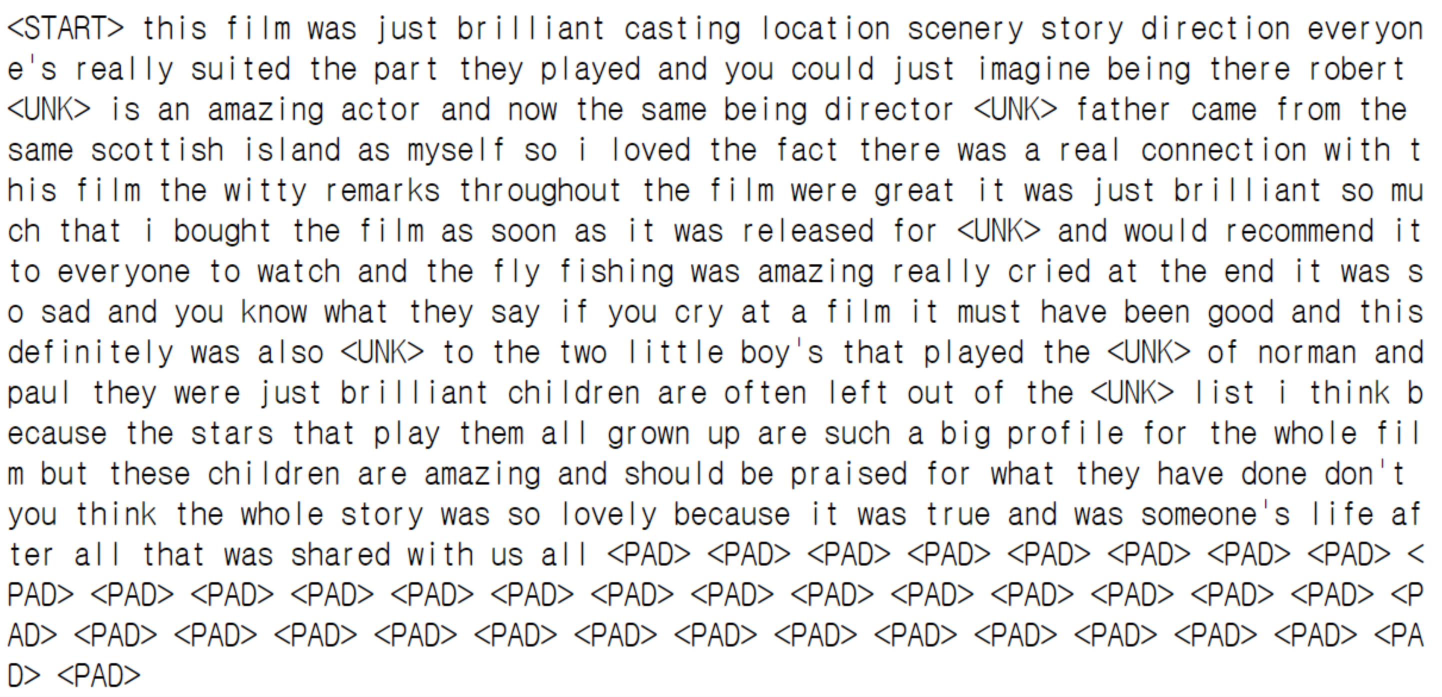
```
(25000, 256)
```



```
word_idx_dict_new['<PAD>']=0
```

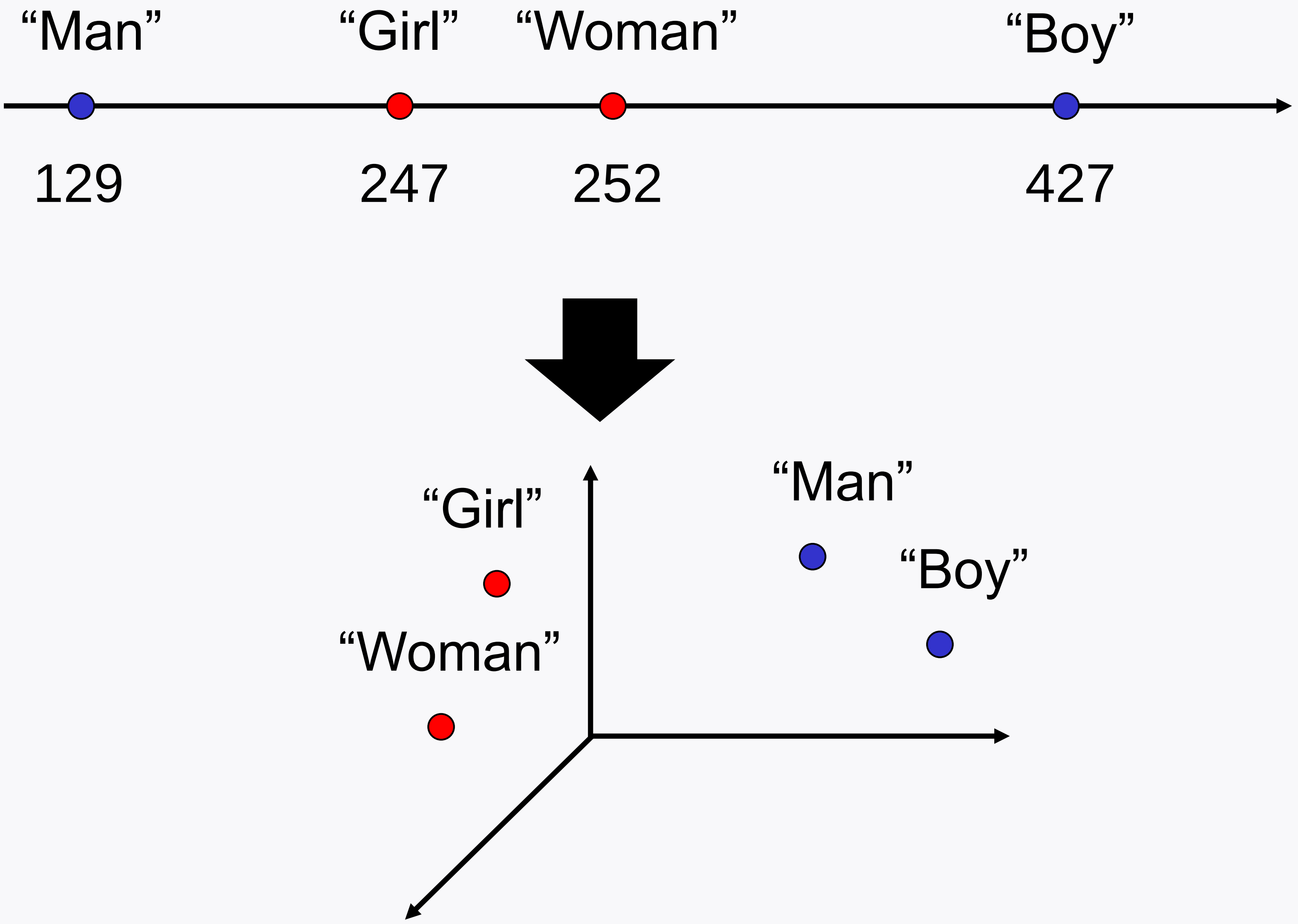


```
print(decode_review(X_train_pad[0]))
```

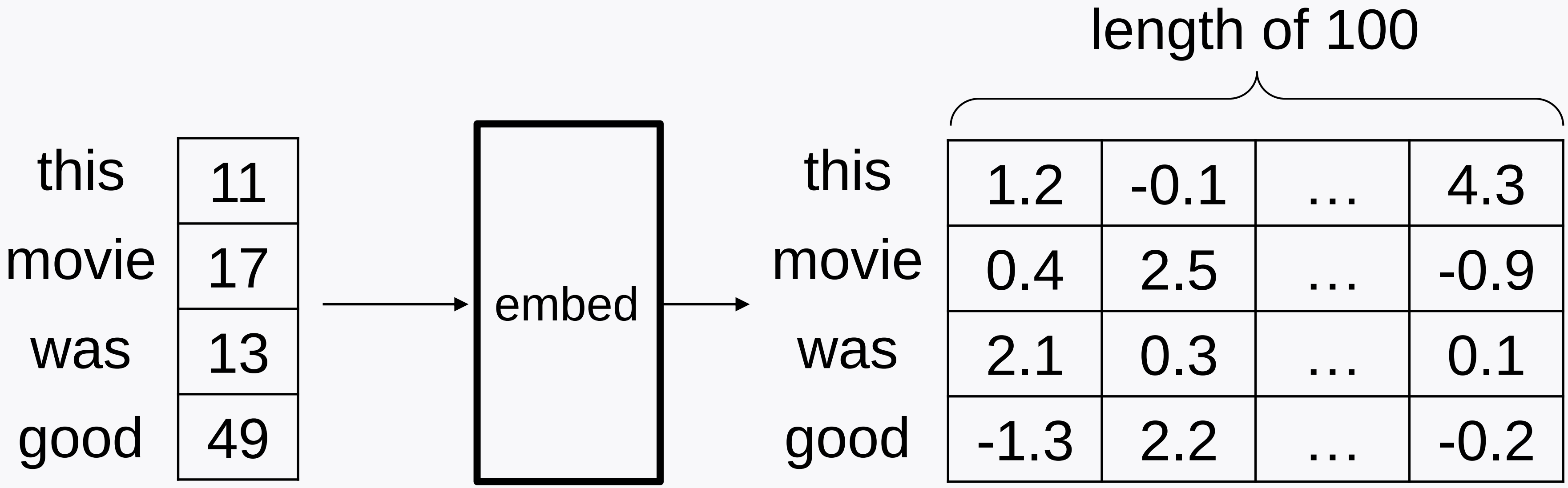


```
<START> this film was just brilliant casting location scenery story direction everyon
e's really suited the part they played and you could just imagine being there robert
<UNK> is an amazing actor and now the same being director <UNK> father came from the
same scottish island as myself so i loved the fact there was a real connection with t
his film the witty remarks throughout the film were great it was just brilliant so mu
ch that i bought the film as soon as it was released for <UNK> and would recommend it
to everyone to watch and the fly fishing was amazing really cried at the end it was s
o sad and you know what they say if you cry at a film it must have been good and this
definitely was also <UNK> to the two little boy's that played the <UNK> of norman and
paul they were just brilliant children are often left out of the <UNK> list i think b
ecause the stars that play them all grown up are such a big profile for the whole fil
m but these children are amazing and should be praised for what they have done don't
you think the whole story was so lovely because it was true and was someone's life af
ter all that was shared with us all <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <
PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <P
AD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PA
D> <PAD>
```

Word embedding



Example



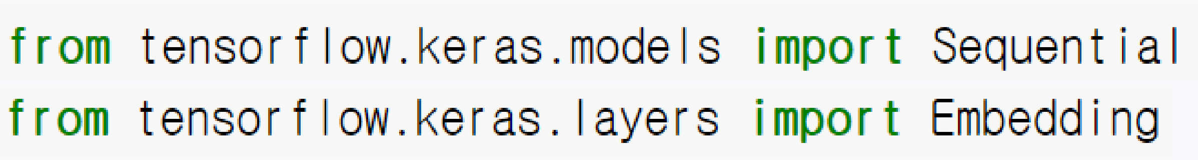
this	11
movie	17
was	13
good	49

embed

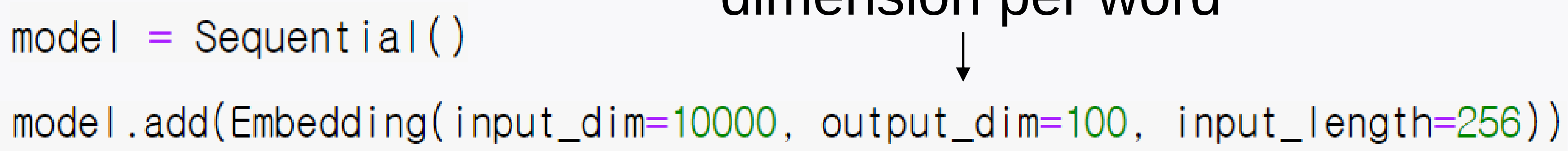
length of 100

this	1.2	-0.1	…	4.3
movie	0.4	2.5	…	-0.9
was	2.1	0.3	…	0.1
good	-1.3	2.2	…	-0.2

Embedding: Tensorflow coding



```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding
```

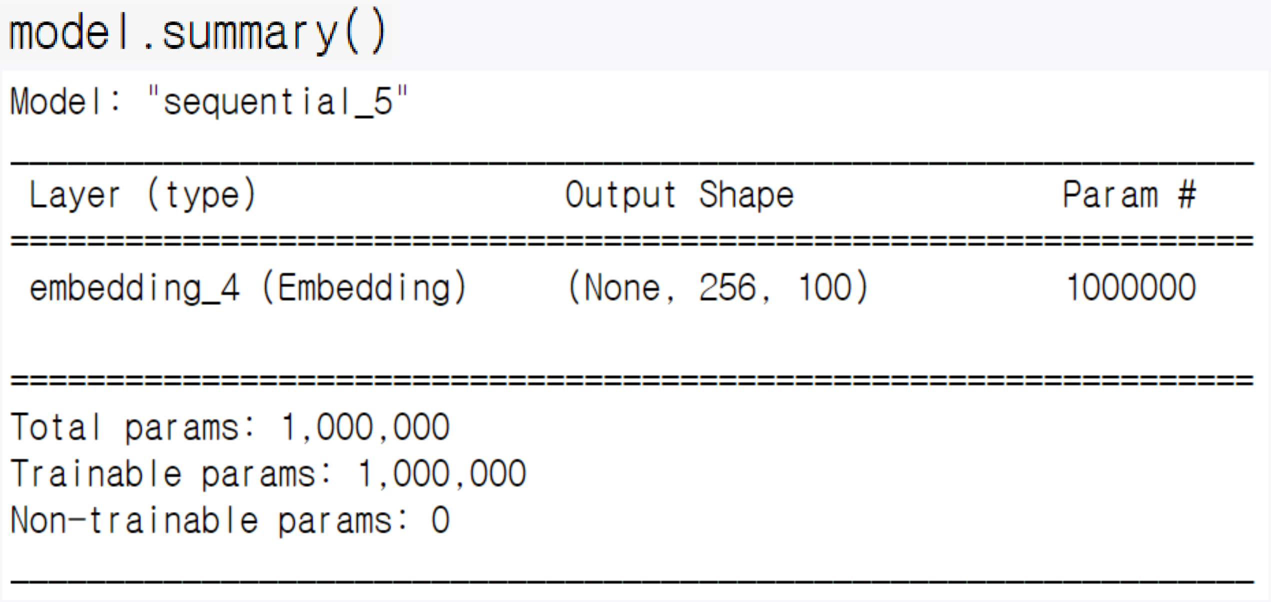


```
model = Sequential()

model.add(Embedding(input_dim=10000, output_dim=100, input_length=256))
```

dimension per word ↓

↑ total number of vocabularies employed



```
model.summary()
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_4 (Embedding)     (None, 256, 100)          1000000

=================================================================
Total params: 1,000,000
Trainable params: 1,000,000
Non-trainable params: 0
_________________________________________________________________
```

Look ahead

Will implement two models:

1. Basic RNN
2. LSTM