In [ ]:
```
#@title Licensed under the Apache License, Version 2.0 (the "Licens
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
# https://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, softwa
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or i
# See the License for the specific language governing permissions a
# limitations under the License.
```

(https://colab.research.google.com/github/lmoroney/dlaicourse/blob/master/TensorF
%20NLP/Course%203%20-%20Week%201%20-%20Lesson%202.ipynb)

Copyright 2019 The TensorFlow Authors.

In [ ]:
```
#@title Licensed under the Apache License, Version 2.0 (the "Licens
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
# https://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, softwa
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or i
# See the License for the specific language governing permissions a
# limitations under the License.
```

In [1]:
```
import tensorflow as tf
from tensorflow import keras


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

sentences = [
    'I love my dog',
    'I love my cat',
    'You love my dog!',
    'Do you think my dog is amazing?'
]

#set an out-of-vocabulary token, so it's used instead of not
#showing up anything altogether
tokenizer = Tokenizer(num_words = 100, oov_token="<OOV>")
tokenizer.fit_on_texts(sentences)
word_index = tokenizer.word_index

#creates an array of the sentences that're tokenized
sequences = tokenizer.texts_to_sequences(sentences)

#just like for images, text too needs to be uniform in
#length, in order to accomplish that we zero-pad the sentences
#to a max length of 5 words (sentences > 5 are truncated)
padded = pad_sequences(sequences, maxlen=5)
print("\nWord Index = " , word_index)
print("\nSequences = " , sequences)
print("\nPadded Sequences:")
print(padded)


# Try with words that the tokenizer wasn't fit to
test_data = [
    'i really love my dog',
    'my dog loves my manatee'
]

#test on sentences that haven't been tokenized by tokenizer
#words included in the sentences that're tokenized take the
#values that you expect them to, new, foreign words take the
#value specified for OOV
test_seq = tokenizer.texts_to_sequences(test_data)
print("\nTest Sequence = ", test_seq)

padded = pad_sequences(test_seq, maxlen=10)
print("\nPadded Test Sequence: ")
print(padded)
```

```
Word Index =  {'<OOV>': 1, 'my': 2, 'love': 3, 'dog': 4, 'i': 5, 'y
t': 7, 'do': 8, 'think': 9, 'is': 10, 'amazing': 11}

Sequences =  [[5, 3, 2, 4], [5, 3, 2, 7], [6, 3, 2, 4], [8, 6, 9, 2
1]]
```

```
Padded Sequences:
[[ 0  5  3  2  4]
 [ 0  5  3  2  7]
 [ 0  6  3  2  4]
 [ 9  2  4 10 11]]

Test Sequence =  [[5, 1, 3, 2, 4], [2, 4, 1, 2, 1]]

Padded Test Sequence:
[[0 0 0 0 0 5 1 3 2 4]
 [0 0 0 0 0 2 4 1 2 1]]
```