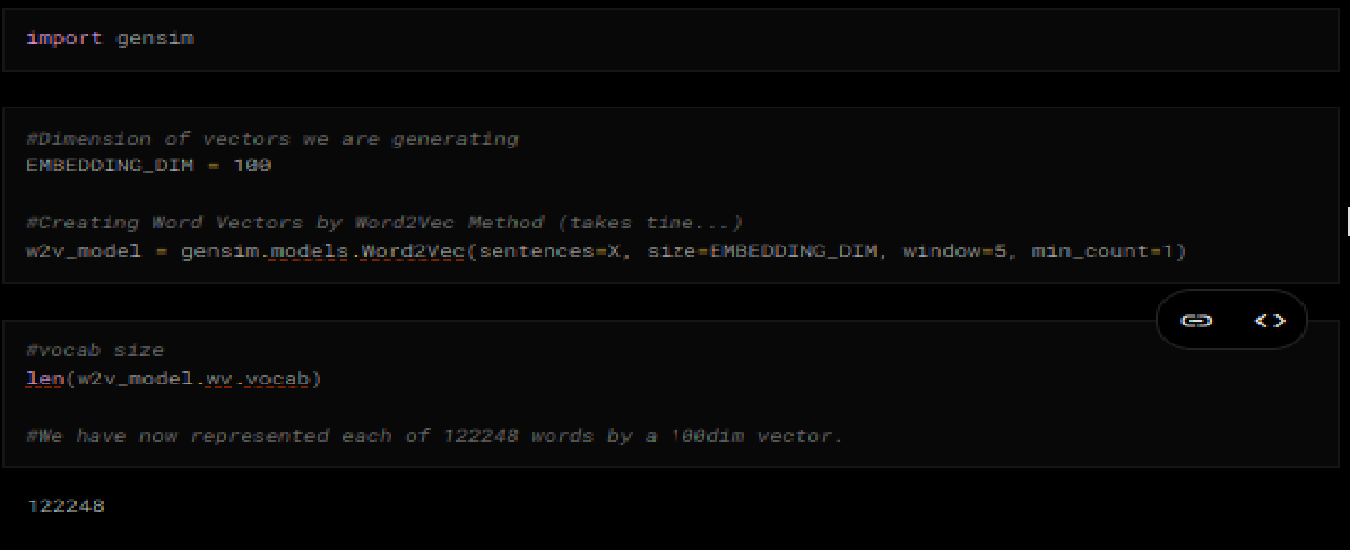
Fake News Detection using NLP

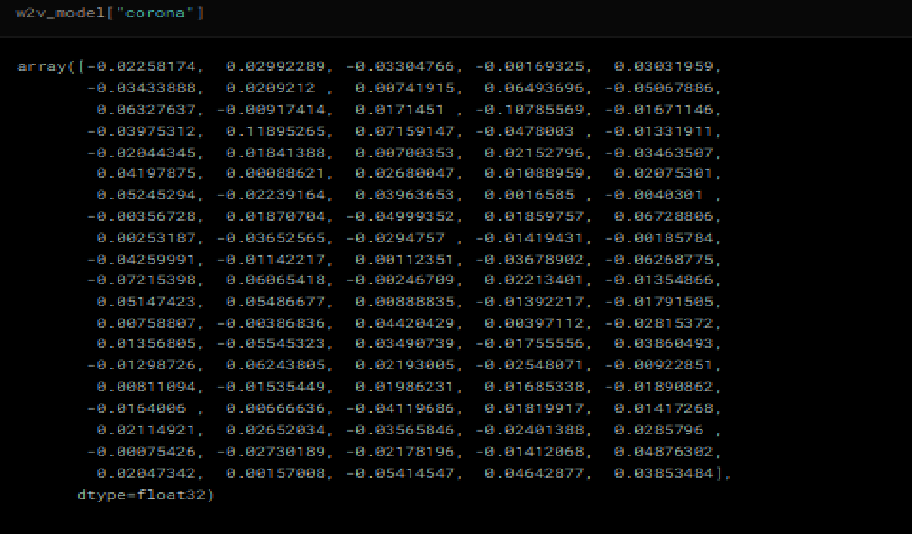
Modeling

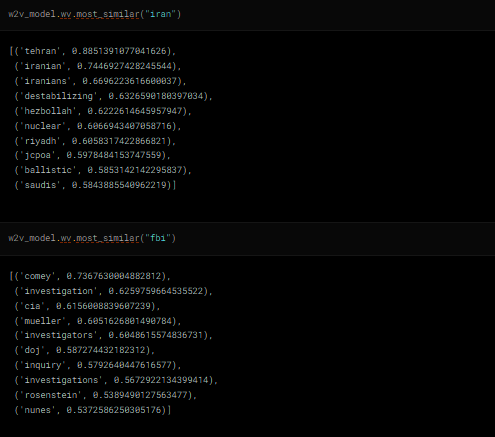
### Vectorization -- Word2Vec

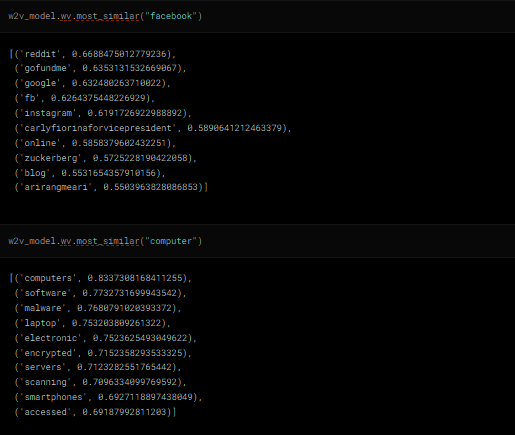
### Word2Vec is one of the most popular technique to learn word embeddings using shallow neural network. Word embedding is the most popular representation of document vocabulary. It is capable of capturing context of a word in a document, semantic and syntactic similarity, relation with other words, etc.

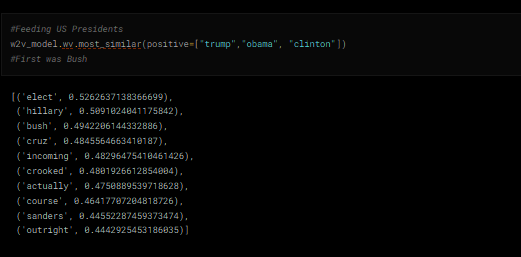


Exploring Vectors #Below is a sample data (Eg: corona)









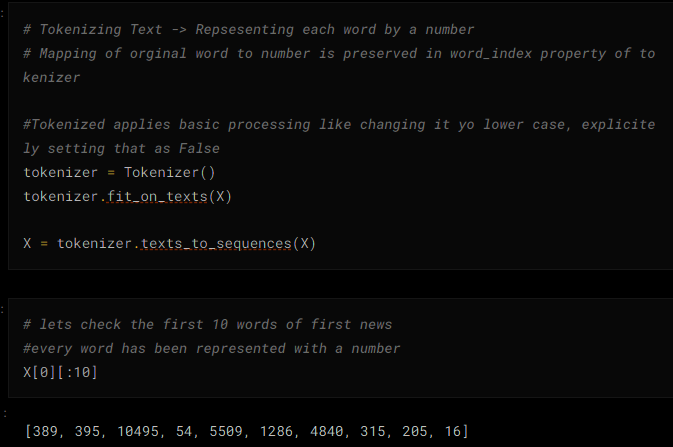
These Vectors will be passed to LSTM/GRU instead of words. 1D-CNN can further be used to extract features from the vectors. Keras has implementation called "**Embedding Layer**" which would create word embeddings(vectors).

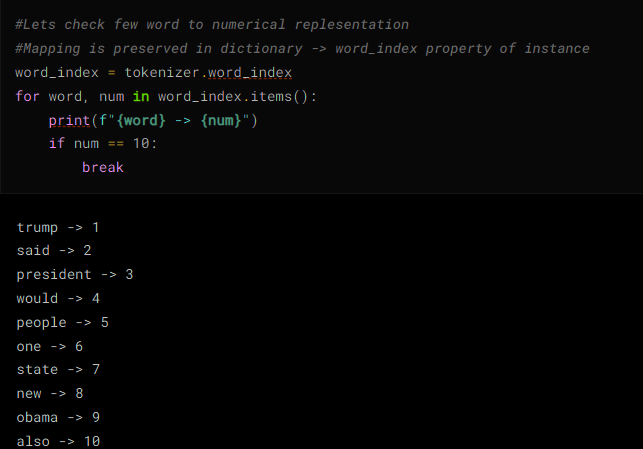
import keras

from keras.preprocessing import text,sequence

from keras.models import Sequential

from keras.layers import Dense,Embedding,LSTM,Dropout

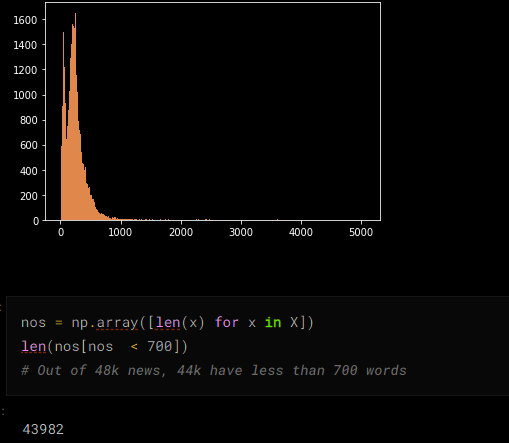




*# For determining size of input...*

plt.hist([len(x) for x **in** X], bins=500)

plt.show()



*#Lets keep all news to 700, add padding to news with less than 700 words and truncating long ones*

maxlen = 700

*#Making all news of size maxlen defined above*

X = pad\_sequences(X, maxlen=maxlen)

*# Adding 1 because of reserved 0 index*

*# Embedding Layer creates one more vector for "UNKNOWN" words, or padded words (0s). This Vector is filled with zeros.*

*# Thus our vocab size inceeases by 1*

vocab\_size = len(tokenizer.word\_index) + 1

*# Function to create weight matrix from word2vec gensim model*

def get\_weight\_matrix(model, vocab):

*# total vocabulary size plus 0 for unknown words*

vocab\_size = len(vocab) + 1

*# define weight matrix dimensions with all 0*

weight\_matrix = np.zeros((vocab\_size, EMBEDDING\_DIM))

*# step vocab, store vectors using the Tokenizer's integer mapping*

for word, i **in** vocab.items():

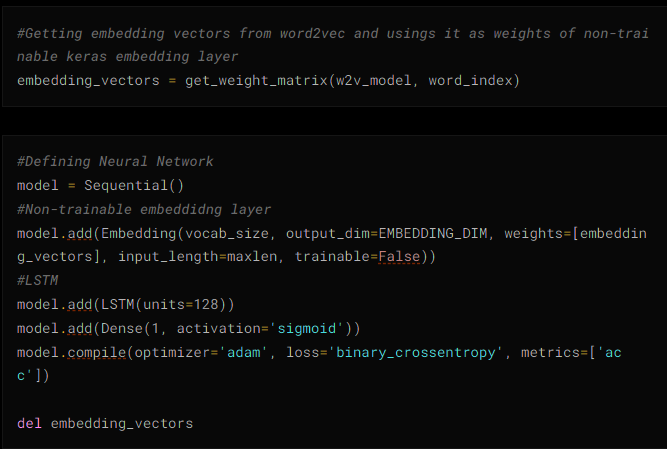
weight\_matrix[i] = model[word]

return weight\_matrix

Create a matrix of mapping between word-index and vectors and use this as weights in embedding layer which accepts numerical-token of word and outputs corresponding vector to inner layer.

It sends vector of zeros to next layer for unknown words which would be tokenized to 0.

Input length of Embedding Layer is the length of each news (700 now due to padding and truncating)



model.summary()

