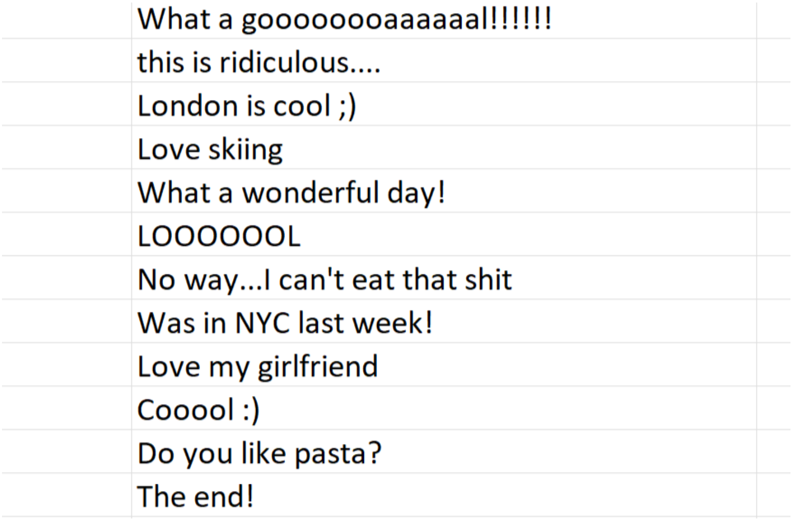
Text Classification of Tweets

By Nicole Joseph

Objective

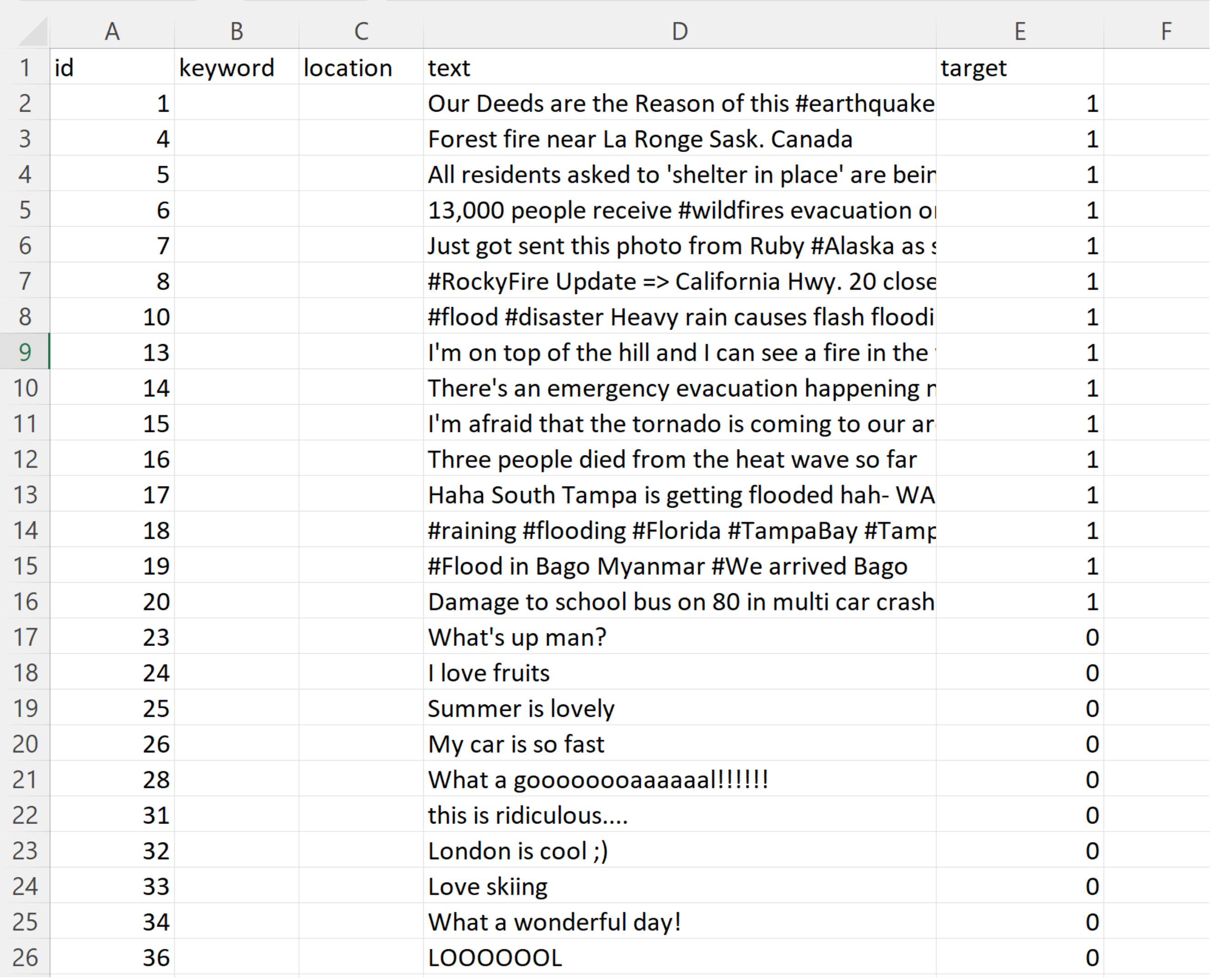
- Implement and train a neural net using Python, TensorFlow, and Keras
- NLP Task: Text classification of tweets as Disaster or No Disaster
- 1 = Disaster
- 0 = No Disaster
- “The average tweet is 28 characters long, with a spike at the 140 character mark.”



What a goooooooaaaaaal!!!!!!
this is ridiculous....
London is cool ;)
Love skiing
What a wonderful day!
LOOOOOOL
No way...I can't eat that shit
Was in NYC last week!
Love my girlfriend
Cooool :)
Do you like pasta?
The end!

The Data Set

- NLP Disaster Tweets from Kaggle
  - Link: https://www.kaggle.com/c/nlp-getting-started
- 7613 total tweets
- 3271 - disaster
- 4342 - no disaster
- Disaster: I know it's a question of interpretation but this is a sign of the apocalypse. I called it https://t.co/my8q1uWIjn
- No Disaster: I'm not gonna lie I'm kinda ready to attack my Senior year ??????????
- 80:20 training testing split



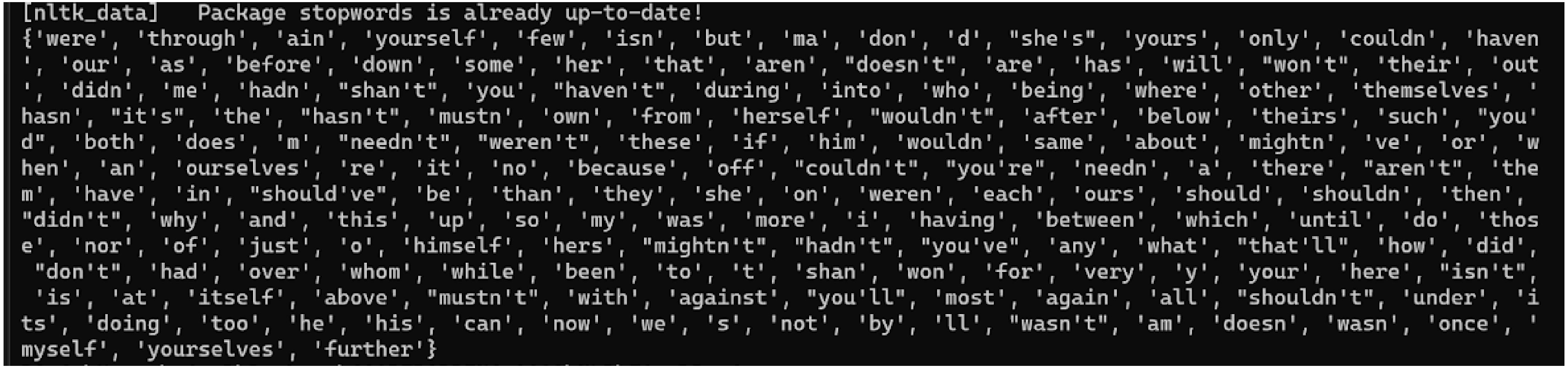
	A	B	C	D	E	F
1	id	keyword	location	text	target	
2	1			Our Deeds are the Reason of this #earthquake	1	
3	4			Forest fire near La Ronge Sask. Canada	1	
4	5			All residents asked to 'shelter in place' are beir	1	
5	6			13,000 people receive #wildfires evacuation or	1	
6	7			Just got sent this photo from Ruby #Alaska as s	1	
7	8			#RockyFire Update => California Hwy. 20 close	1	
8	10			#flood #disaster Heavy rain causes flash floodi	1	
9	13			I'm on top of the hill and I can see a fire in the	1	
10	14			There's an emergency evacuation happening r	1	
11	15			I'm afraid that the tornado is coming to our ar	1	
12	16			Three people died from the heat wave so far	1	
13	17			Haha South Tampa is getting flooded hah- WA	1	
14	18			#raining #flooding #Florida #TampaBay #Tamp	1	
15	19			#Flood in Bago Myanmar #We arrived Bago	1	
16	20			Damage to school bus on 80 in multi car crash	1	
17	23			What's up man?	0	
18	24			I love fruits	0	
19	25			Summer is lovely	0	
20	26			My car is so fast	0	
21	28			What a goooooooaaaaaal!!!!!!	0	
22	31			this is ridiculous....	0	
23	32			London is cool ;)	0	
24	33			Love skiing	0	
25	34			What a wonderful day!	0	
26	36			LOOOOOOL	0	

Preprocessing

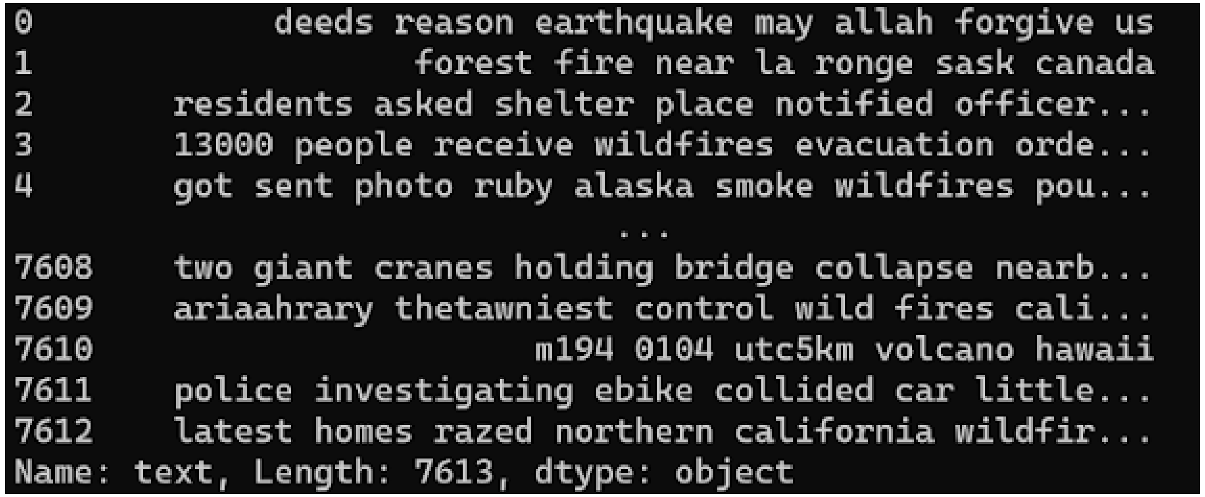
- Remove URL using regular expressions
- Remove punctuation
- !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
- “You can deduce that the **intensity** of a particular communication is high by the amount of exclamation marks used, which could be an indication of a strong positive or negative emotion”

Preprocessing

- Remove stopwords using the nltk toolkit



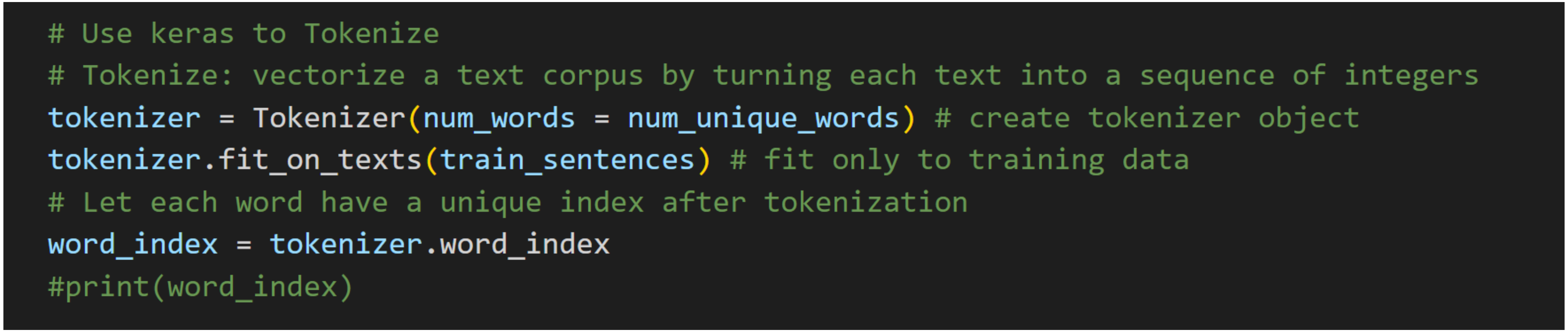
```
[nltk_data]   Package stopwords is already up-to-date!
{'were', 'through', 'ain', 'yourself', 'few', 'isn', 'but', 'ma', 'don', 'd', "she's", 'yours', 'only', 'couldn', 'haven
', 'our', 'as', 'before', 'down', 'some', 'her', 'that', 'aren', "doesn't", 'are', 'has', 'will', "won't", 'their', 'out
', 'didn', 'me', 'hadn', "shan't", 'you', "haven't", 'during', 'into', 'who', 'being', 'where', 'other', 'themselves', '
hasn', "it's", 'the', "hasn't", 'mustn', 'own', 'from', 'herself', "wouldn't", 'after', 'below', 'theirs', 'such', "you'
d", 'both', 'does', 'm', "needn't", "weren't", 'these', 'if', 'him', 'wouldn', 'same', 'about', 'mightn', 've', 'or', 'w
hen', 'an', 'ourselves', 're', 'it', 'no', 'because', 'off', "couldn't", "you're", 'needn', 'a', 'there', "aren't", 'the
m', 'have', 'in', "should've", 'be', 'than', 'they', 'she', 'on', 'weren', 'each', 'ours', 'should', 'shouldn', 'then',
"didn't", 'why', 'and', 'this', 'up', 'so', 'my', 'was', 'more', 'i', 'having', 'between', 'which', 'until', 'do', 'thos
e', 'nor', 'of', 'just', 'o', 'himself', 'hers', "mightn't", "hadn't", "you've", 'any', 'what', "that'll", 'how', 'did',
 "don't", 'had', 'over', 'whom', 'while', 'been', 'to', 't', 'shan', 'won', 'for', 'very', 'y', 'your', 'here', "isn't",
 'is', 'at', 'itself', 'above', "mustn't", 'with', 'against', "you'll", 'most', 'again', 'all', "shouldn't", 'under', 'i
ts', 'doing', 'too', 'he', 'his', 'can', 'now', 'we', 's', 'not', 'by', 'll', "wasn't", 'am', 'doesn', 'wasn', 'once', '
myself', 'yourselves', 'further'}
```



```
0             deeds reason earthquake may allah forgive us
1                    forest fire near la ronge sask canada
2        residents asked shelter place notified officer...
3        13000 people receive wildfires evacuation orde...
4        got sent photo ruby alaska smoke wildfires pou...
                               ...
7608     two giant cranes holding bridge collapse nearb...
7609     ariaahrary thetawniest control wild fires cali...
7610                     m194 0104 utc5km volcano hawaii
7611     police investigating ebike collided car little...
7612     latest homes razed northern california wildfir...
Name: text, Length: 7613, dtype: object
```

Tokenization

- Count unique words
- Use the Counter object to iterate over each line in the text column and split into an array of words
- Output for len(counter): 17971 unique words



```
# Use keras to Tokenize
# Tokenize: vectorize a text corpus by turning each text into a sequence of integers
tokenizer = Tokenizer(num_words = num_unique_words) # create tokenizer object
tokenizer.fit_on_texts(train_sentences) # fit only to training data
# Let each word have a unique index after tokenization
word_index = tokenizer.word_index
#print(word_index)
```

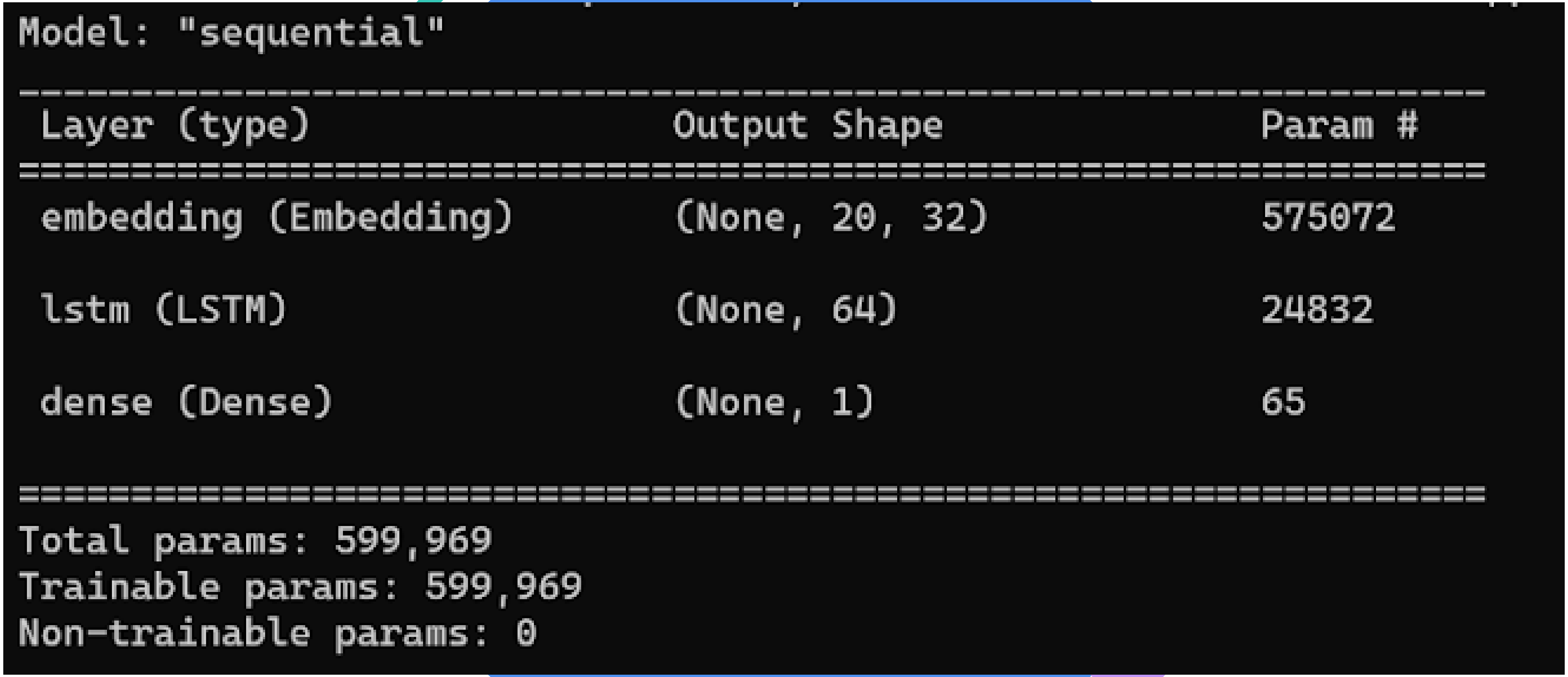
Tokenization

- Use zero padding to ensure that all sequences have the same length
- Set max number of words in sequence = 20
  - Max length was set to 20 because that seemed fitting for a tweet. This number can be played around with but when it's a larger number, the training might take longer
  - This max length will be set as the input length to the embedding layer of the training model

three people died heat wave far

[520, 8, 395, 156, 297, 411]

[520 8 395 156 297 411 0 0 0 0 0 0 0 0 0 0 0 0 0 0]



```
Model: "sequential"
-----------------------------------------------------------------
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 20, 32)            575072

 lstm (LSTM)                 (None, 64)                24832

 dense (Dense)               (None, 1)                 65

=================================================================
Total params: 599,969
Trainable params: 599,969
Non-trainable params: 0
```

Model Summary

- Sequential Model with Embedding layer, LSTM layer, and a Dense layer
- Word embeddings turn positive integers (indexes) into dense vectors of fixed size.
    - (other NLP approaches could be one-hot-encoding)
- Word embeddings give us a way to use an efficient, dense representation in which similar words have a similar encoding. Importantly, you do not have to specify this encoding by hand
- LSTM (Long Short-Term Memory) that are a variety of recurrent neural networks (RNNs) capable of learning long-term dependencies, especially in sequence prediction problems
- Dense layer with sigmoid activation function since a 0 or 1 classification is desired

Thank you!