Autoimmune Tweets using Lemmatized Tweets with 8 categories of autoimmune diseases¶

Those being: 1:Multiple Sclerosis, 2:Celiac, 3: Leukemia, 4: Hashimoto, 5: Fibromyalgia, 6: Kidney Disease, 7: Rheumatoid Arthritis, 8: Chron's Disease

Tweets were taken from respective diseases in early December 2019 from 13 to 119 tweets for each disease, as many as were found that weren't mostly marketing, using '' treatment' in the search

In [1]:

```
%matplotlib inline

import pandas as pd

import matplotlib.pyplot as plt

from textblob import TextBlob

import sklearn

import numpy as np

from sklearn.feature_extraction.text import CountVectorizer

from sklearn.naive_bayes import MultinomialNB

from sklearn.metrics import classification_report, f1_score, accuracy_score, confusion
_matrix


np.random.seed(507)
```

In [2]:

```
reviews = pd.read_csv('TargetReady.csv', encoding = 'unicode_escape')

#the encoding needed for python3 handling nonASCII chars
```

In [3]:

```
reviews.head()
```

Out[3]:

	Tweet	Type
0	UNKNOWNResearchCa\r\n@UNKNOWN_ARC\r\n·\r\n19h\...	Rheumatoid Arthritis
1	UNKNOWNatology Advisor\r\n@UNKNOWNAdvisor\r\n·...	Rheumatoid Arthritis
2	UNKNOWN Community\r\n@our_UNKNOWN\r\n·\r\nDec ...	Rheumatoid Arthritis
3	UNKNOWN National Research Foundation\r\n@CureU...	Rheumatoid Arthritis
4	Orthopedic News\r\n@Orthopedics_Bio\r\n·\r\nDe...	Rheumatoid Arthritis

In [4]:

```
reviews.tail()
```

Out[4]:

	Tweet	Type
502	All Ezine\r\n@allezine\r\n·\r\nJun 13, 2011\r\...	Chron's Disease
503	Brian Coombes\r\n@BrianKCoombes\r\n·\r\nSep 6\...	Chron's Disease
504	Purpose ?\r\n@HappyBelieber\r\n·\r\nJan 19, 20...	Chron's Disease
505	K. Ketels-Lichtig\r\n@kklichtig\r\n·\r\nOct 25...	Chron's Disease
506	-DC-™\r\n @FuckwitdaDC\r\n·\r\nJul 8, 2015\r\...	Chron's Disease

In [5]:

```
reviews.shape
```

Out[5]:

```
(507, 2)
```

In [6]:

```
reviews = reviews.reindex(np.random.permutation(reviews.index))


print(reviews.head())
print(reviews.tail())
```

```
                                                 Tweet                Type
288  Aleksandar dr Petrov\r\n@aleksandar_BG\r\n·\r\...  Multiple Sclerosis
70   Beyond UNKNOWN\r\n@BeyondUNKNOWN\r\n·\r\nSep 1...      Celiac Disease
184  #HandsOffVenezuela\r\n@ChicoFreedom\r\n·\r\nDe...             Leukemia
459  Adult & Pediatric Ear, Nose & Throat\r\n@EarAd...   Hashimoto Disease
448  Angela J. White\r\n@50Plushealths\r\n·\r\nDec ...        Fibromyalgia
                                                 Tweet                Type
136  CURE Magazine\r\n@cure_magazine\r\n·\r\nDec 3\...             Leukemia
503  Brian Coombes\r\n@BrianKCoombes\r\n·\r\nSep 6\...     Chron's Disease
295  Glynis Edwards\r\n@Glynis4B12\r\n·\r\nNov 26\r...  Multiple Sclerosis
452  Mavz\r\n@mattymavz\r\n·\r\nNov 5, 2018\r\nIt's...        Fibromyalgia
112  GrupoCronosSEFH\r\n@GRUPOCRONOSSEF1\r\n·\r\nDe...      Kidney Disease
```

In [7]:

```
reviews.groupby('Type').describe()
```

Out[7]:

	Tweet			
	count	unique	top	freq
Type				
Celiac Disease	50	50	Truthbetold?\r\n@wlkthlne\r\n·\r\nNov 30\r\nRe...	1
Chron's Disease	19	19	-DC-™\r\n @FuckwitdaDC\r\n·\r\nJul 8, 2015\r\...	1
Fibromyalgia	99	96	Women In Pain\r\n@forgrace\r\n·\r\nNov 26\r\nF...	2
Hashimoto Disease	30	29	Colorado Natural Med\r\n@drgravesCO\r\n·\r\nDe...	2
Kidney Disease	43	43	B.K. Arogyam\r\n@KArogyam\r\n·\r\nDec 2\r\nIf ...	1
Leukemia	119	119	Sabrcare Trust\r\n@sabrcaretrust\r\n·\r\nDec 2...	1
Multiple Sclerosis	119	119	Multiple Sclerosis\r\n@UNKNOWN_Bio\r\n·\r\nDec...	1
Rheumatoid Arthritis	28	28	Frontiers Medicine\r\n@FrontMedicine\r\n·\r\nO...	1

In [8]:

```
reviews['length'] = reviews['Tweet'].map(lambda text: len(text))

print(reviews.head())
```

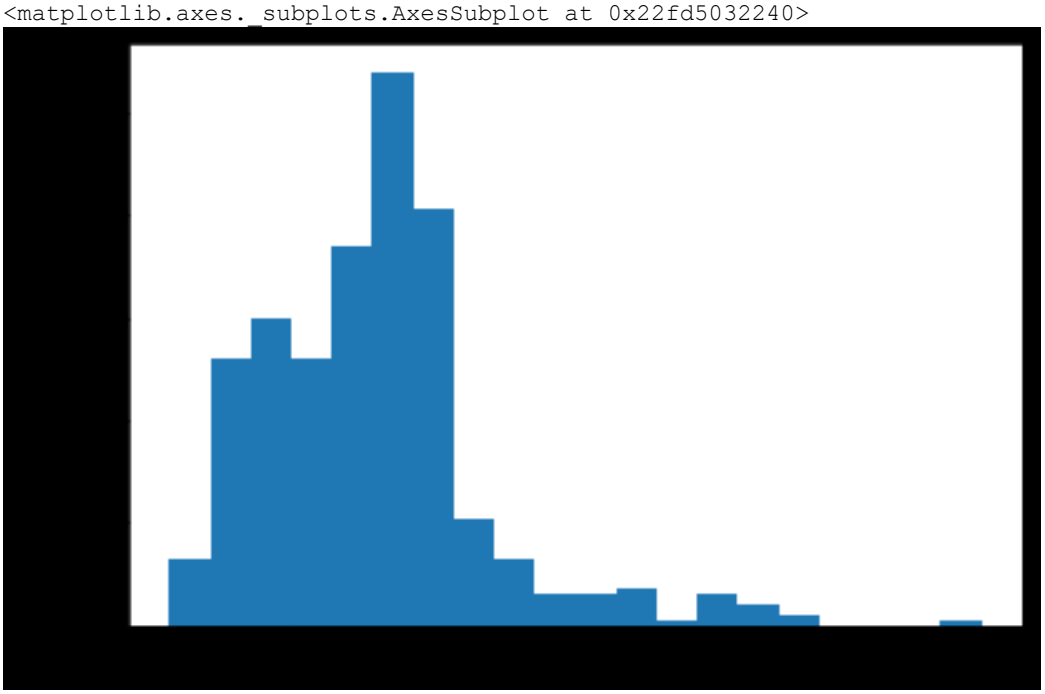
```
                                                 Tweet                 Type  \
288  Aleksandar dr Petrov\r\n@aleksandar_BG\r\n·\r\...  Multiple Sclerosis
70   Beyond UNKNOWN\r\n@BeyondUNKNOWN\r\n·\r\nSep 1...      Celiac Disease
184  #HandsOffVenezuela\r\n@ChicoFreedom\r\n·\r\nDe...           Leukemia
459  Adult & Pediatric Ear, Nose & Throat\r\n@EarAd...   Hashimoto Disease
448  Angela J. White\r\n@50Plushealths\r\n·\r\nDec ...        Fibromyalgia

     length
288     281
70      247
184     317
459     142
448     255
```

In [9]:

```
reviews.length.plot(bins=20, kind='hist')
```

Out[9]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x22fd5032240>
```



In [10]:

```
reviews.length.describe()
```

Out[10]:

```
count    507.000000
mean     276.532544
std      104.546869
min       87.000000
```

```
25%      201.500000
50%      279.000000
75%      320.000000
max      847.000000
Name: length, dtype: float64
```

In [11]:

```
print(list(reviews.Tweet[reviews.length > 700].index)) #near the max for length of Lem
matizedTweets

print(list(reviews.Type[reviews.length > 700]))

print(list(reviews.Tweet[reviews.length > 700]))
```

```
[151]
['Leukemia']
['BTS Mauritius\r\n@BTSMauritius\r\n·\r\n19h\r\nThe Korea Leukemia Children\x92s Found
ation announced that 553 ARMYs donated blood in honour of @BTS_twt\r\n\x92s Jin birthd
ay!\r\n\r\nIt\x92ll be used for children with cancer who need to receive large blood t
ransfusions during treatment, helping to ease the burden of costs for patients\x92 fam
ilies.\r\nQuote Tweet\r\n??????\r\n??\r\n??\r\n@_nojam_nolife\r\n · Dec 3\r\n????? ? ?
?, ?? ?? ??? ??(??)  \r\n??\r\nhttp://entertain.v.daum.net/v/20191204091525391\r\n\r\n
#????? #BTS @BTS_twt \r\n\r\n??????????? ?????? ????? ? ????? ?? ?? ??? 12? 4? ????? ?
? ???? ?? ??? ???? ??? 553?? ????? ???.\r\nImage\r\nImage\r\nImage\r\nImage\r\nImpriso
ned Babies\r\n@aptlmetin\r\n·\r\nDec 3\r\nAkif Acute Lymphoblastic Leukemia patient. H
e is going through a heavy treatment process. He needs medicare. Let mom?enay DA?TAN h
ave her trial without arrest!\r\n#InternationalDisabilityDay ']
```

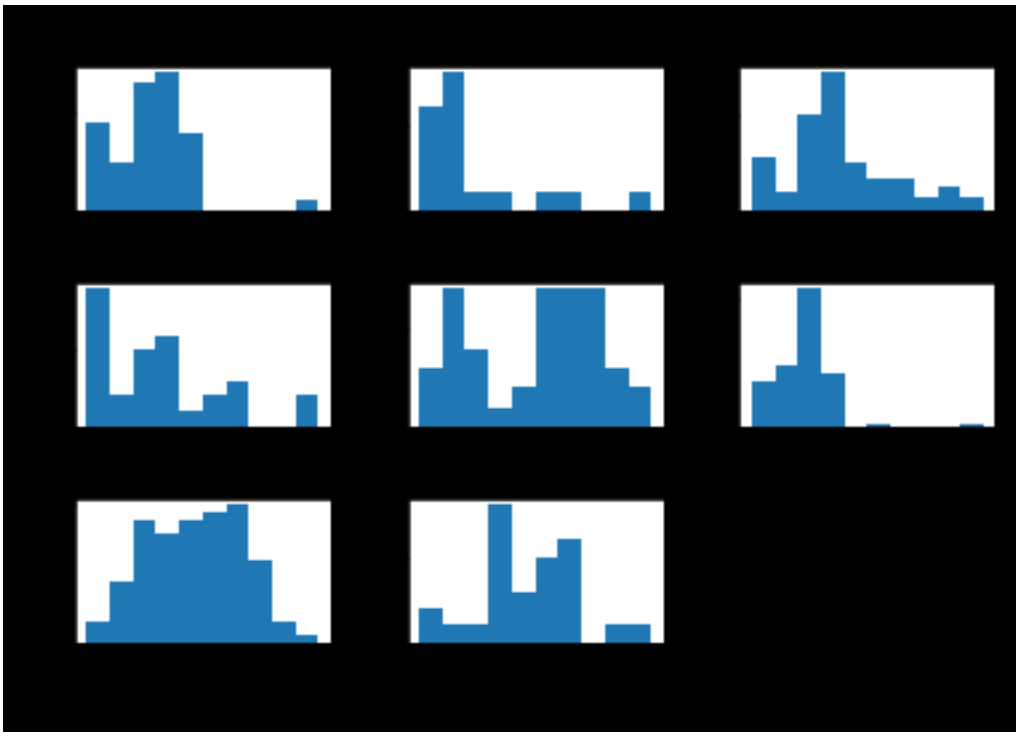
In [12]:

```
%%time

reviews.hist(column='length', by='Type', bins=10)
```

```
Wall time: 484 ms
```

Out[12]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000022FD511BD30>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000022FD51796D8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000022FD51A4C88>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x0000022FD51E3278>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000022FD5213828>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000022FD5246DD8>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x0000022FD52843C8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000022FD52B39B0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000022FD52B39E8>]],
      dtype=object)
```



In [13]:

```
def split_into_tokens(review):


    #review = unicode(review, 'iso-8859-1')# in python 3 the default of str() previous
ly python2 as unicode() is utf-8
    return TextBlob(review).words
```

In [14]:

```
reviews.Tweet.head().apply(split_into_tokens)
```

Out[14]:

```
288    [Aleksandar, dr, Petrov, aleksandar_BG, ·, Dec...
70     [Beyond, UNKNOWN, BeyondUNKNOWN, ·, Sep, 17, 2...
184    [HandsOffVenezuela, ChicoFreedom, ·, Dec, 2, o...
459    [Adult, Pediatric, Ear, Nose, Throat, EarAdult...
448    [Angela, J, White, 50Plushealths, ·, Dec, 15, ...
Name: Tweet, dtype: object
```

In [15]:

```
TextBlob("hello world, how is it going?").tags
```

Out[15]:

```
[('hello', 'JJ'),
 ('world', 'NN'),
 ('how', 'WRB'),
 ('is', 'VBZ'),
 ('it', 'PRP'),
```

```
 ('going', 'VBG')]
```

In [16]:

```
import nltk

nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\m\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[16]:

```
True
```

In [17]:

```
from nltk.corpus import stopwords


stop = stopwords.words('english')

stop = stop + [u'a',u'b',u'c',u'd',u'e',u'f',u'g',u'h',u'i',u'j',u'k',u'l',u'm',u'n',u
'o',u'p',u'q',u'r',u's',u't',u'v',u'w',u'x',u'y',u'z']
```

In [18]:

```
def split_into_lemmas(review):

    #review = unicode(review, 'iso-8859-1')

    review = review.lower()

    #review = unicode(review, 'utf8').lower()

    #review = str(review).lower()

    words = TextBlob(review).words

    # for each word, take its "base form" = lemma

    return [word.lemma for word in words if word not in stop]


reviews.Tweet.head().apply(split_into_lemmas)
```

Out[18]:

```
288    [aleksandar, dr, petrov, aleksandar_bg, ·, dec...
70     [beyond, unknown, beyondunknown, ·, sep, 17, 2...
184    [handsoffvenezuela, chicofreedom, ·, dec, 2, o...
459    [adult, pediatric, ear, nose, throat, earadult...
448    [angela, white, 50plushealths, ·, dec, 15, 201...
Name: Tweet, dtype: object
```

In [19]:

```
%%time

bow_transformer = CountVectorizer(analyzer=split_into_lemmas, ngram_range=(1,3)).fit(r
eviews['Tweet'])

print(len(bow_transformer.vocabulary_))
```

```
4791
Wall time: 1.05 s
```

In [20]:

```
bow_transformer
```

Out[20]:

```
CountVectorizer(analyzer=<function split_into_lemmas at 0x0000022FCC9CC730>,
                binary=False, decode_error='strict',
                dtype=<class 'numpy.int64'>, encoding='utf-8', input='content',
                lowercase=True, max_df=1.0, max_features=None, min_df=1,
                ngram_range=(1, 3), preprocessor=None, stop_words=None,
                strip_accents=None, token_pattern='(?u)\\b\\w\\w+\\b',
                tokenizer=None, vocabulary=None)
```

In [21]:

```
review4 = reviews['Tweet'][148]

print(review4)
```

```
Peking University
@PKU1898
·
Nov 30
Published in Cold Spring Harbor Perspectives in Medicine, #Peking University□s Wu Hong
and team analyzed connections between a tumor suppressing gene called PTEN, the format
ion of blood cell components, and leukemia. #PekingScience
```

In [22]:

```
bow4 = bow_transformer.transform([review4])

print(bow4)
```

```
  (0, 113)       1
  (0, 371)       1
  (0, 705)       1
  (0, 816)       1
  (0, 898)       1
  (0, 1023)      1
  (0, 1062)      1
  (0, 1085)      1
  (0, 1779)      1
  (0, 1847)      1
  (0, 1991)      1
  (0, 2096)      1
  (0, 2546)      1
  (0, 2789)      1
  (0, 3112)      1
  (0, 3304)      2
  (0, 3305)      1
  (0, 3331)      1
  (0, 3370)      1
  (0, 3534)      1
  (0, 3536)      1
  (0, 4024)      1
  (0, 4142)      1
  (0, 4207)      1
  (0, 4403)      1
  (0, 4474)      1
```

```
  (0, 4476)      1
  (0, 4728)      1
  (0, 4789)      1
```

In [23]:

```
%%time

reviews_bow = bow_transformer.transform(reviews['Tweet'])

print('sparse matrix shape:', reviews_bow.shape)

print('number of non-zeros:', reviews_bow.nnz)

print('sparsity: %.2f%%' % (100.0 * reviews_bow.nnz / (reviews_bow.shape[0] * reviews_
bow.shape[1])))
```

```
sparse matrix shape: (507, 4791)
number of non-zeros: 12992
sparsity: 0.53%
Wall time: 1.02 s
```

In [24]:

```
# Split/splice into training ~ 80% and testing ~ 20%

reviews_bow_train = reviews_bow[:400]

reviews_bow_test = reviews_bow[400:]

reviews_sentiment_train = reviews['Type'][:400]

reviews_sentiment_test = reviews['Type'][400:]


print(reviews_bow_train.shape)

print(reviews_bow_test.shape)
```

```
(400, 4791)
(107, 4791)
```

In [25]:

```
%time review_sentiment = MultinomialNB().fit(reviews_bow_train, reviews_sentiment_trai
n)
```

```
Wall time: 15.6 ms
```

In [26]:

```
print('predicted:', review_sentiment.predict(bow4)[0])

print('expected:', reviews.Type[151])
```

```
predicted: Leukemia
expected: Leukemia
```

In [27]:

```
predictions = review_sentiment.predict(reviews_bow_test)

print(predictions)
```

```
['Fibromyalgia' 'Multiple Sclerosis' 'Leukemia' 'Leukemia' 'Leukemia'
 'Fibromyalgia' 'Kidney Disease' 'Multiple Sclerosis'
 'Rheumatoid Arthritis' 'Multiple Sclerosis' 'Multiple Sclerosis'
 'Multiple Sclerosis' 'Fibromyalgia' 'Fibromyalgia' 'Leukemia'
 'Hashimoto Disease' 'Fibromyalgia' 'Fibromyalgia' 'Fibromyalgia'
 'Multiple Sclerosis' 'Leukemia' 'Kidney Disease' 'Multiple Sclerosis'
 'Multiple Sclerosis' 'Celiac Disease' 'Fibromyalgia' 'Fibromyalgia'
 'Fibromyalgia' 'Fibromyalgia' 'Fibromyalgia' 'Hashimoto Disease'
 'Fibromyalgia' 'Celiac Disease' 'Multiple Sclerosis' 'Multiple Sclerosis'
 'Leukemia' 'Leukemia' 'Leukemia' 'Fibromyalgia' 'Fibromyalgia'
 'Multiple Sclerosis' 'Fibromyalgia' 'Multiple Sclerosis' 'Leukemia'
 'Multiple Sclerosis' 'Leukemia' 'Multiple Sclerosis' 'Leukemia'
 'Leukemia' 'Multiple Sclerosis' 'Leukemia' 'Hashimoto Disease'
 'Multiple Sclerosis' 'Multiple Sclerosis' 'Multiple Sclerosis' 'Leukemia'
 'Leukemia' 'Fibromyalgia' 'Multiple Sclerosis' 'Hashimoto Disease'
 'Leukemia' 'Leukemia' 'Leukemia' 'Leukemia' 'Multiple Sclerosis'
 'Fibromyalgia' 'Hashimoto Disease' 'Fibromyalgia' 'Fibromyalgia'
 'Leukemia' 'Multiple Sclerosis' 'Fibromyalgia' 'Celiac Disease'
 'Celiac Disease' 'Celiac Disease' 'Multiple Sclerosis'
 'Multiple Sclerosis' 'Leukemia' 'Fibromyalgia' 'Leukemia' 'Fibromyalgia'
 'Multiple Sclerosis' 'Fibromyalgia' 'Leukemia' 'Leukemia'
 'Multiple Sclerosis' 'Fibromyalgia' 'Fibromyalgia' 'Leukemia'
 'Fibromyalgia' 'Multiple Sclerosis' 'Multiple Sclerosis'
 'Hashimoto Disease' 'Fibromyalgia' 'Fibromyalgia' 'Leukemia'
 'Multiple Sclerosis' 'Multiple Sclerosis' 'Fibromyalgia' 'Celiac Disease'
 'Multiple Sclerosis' 'Leukemia' 'Leukemia' 'Rheumatoid Arthritis'
 'Multiple Sclerosis' 'Fibromyalgia' 'Multiple Sclerosis']
```

In [28]:

```
print('accuracy', accuracy_score(reviews_sentiment_test, predictions))

print('confusion matrix\n', confusion_matrix(reviews_sentiment_test, predictions))

print('(row=expected, col=predicted)')
```

```
accuracy 0.7663551401869159
confusion matrix
 [[ 2  0  2  0  0  0  2  0]
 [ 1  0  0  0  0  0  0  2]
 [ 0  0 22  0  0  0  1  0]
 [ 1  0  1  6  0  0  0  0]
 [ 1  0  2  0  1  0  2  0]
 [ 0  0  1  0  0 28  0  0]
 [ 1  0  2  0  1  0 23  0]
 [ 0  0  1  0  0  0  4  0]]
(row=expected, col=predicted)
```

In [29]:

```
print(classification_report(reviews_sentiment_test, predictions))

#The F1 score can be interpreted as a weighted average of the precision and recall,

#where an F1 score reaches its best value at 1 and worst score at 0.
```

```
                    precision    recall  f1-score   support

      Celiac Disease       0.33      0.33      0.33         6
     Chron's Disease       0.00      0.00      0.00         3
        Fibromyalgia       0.71      0.96      0.81        23
   Hashimoto Disease       1.00      0.75      0.86         8
      Kidney Disease       0.50      0.17      0.25         6
```

```
            Leukemia       1.00      0.97      0.98        29
  Multiple Sclerosis       0.72      0.85      0.78        27
Rheumatoid Arthritis       0.00      0.00      0.00         5

            accuracy                           0.77       107
           macro avg       0.53      0.50      0.50       107
        weighted avg       0.73      0.77      0.73       107

c:\users\m\anaconda2\envs\python36\lib\site-packages\sklearn\metrics\classification.py
:1437: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to
0.0 in labels with no predicted samples.
  'precision', 'predicted', average, warn_for)
```

In [77]:

```
def predict_review(new_review):

    new_sample = bow_transformer.transform([new_review])

    p = np.around(review_sentiment.predict_proba(new_sample), decimals=2)

    print(new_review,p,'\tmax:',np.max(p))
```

The respective probabilities correspond to those diseases alphebatized as [[1-Celiac Disease, 2-Chron's Disease, 3-Fibromyalgia, 4-Hashimoto, 5-Kidney Disease, 6-Leukemia, 7-Multiple Sclerosis, 8-Rheumatoid Arthritis]]

In [78]:

```
predict_review('driving to the hospital.')


predict_review('When is lunch?')


predict_review('Theme parks are great.')

predict_review('Working is great if it pays the bills.')

#a snippet of an actual tweet from RA

predict_review('Treatment broadspectrum betalactam antibiotics including sulfonamide t
rimethoprim associated diagnosis.')
```

```
driving to the hospital. [[0.06 0.08 0.13 0.03 0.1  0.4  0.17 0.04]]        max: 0.4
When is lunch? [[0.11 0.04 0.19 0.06 0.09 0.22 0.23 0.06]] max: 0.23
Theme parks are great. [[0.14 0.03 0.15 0.04 0.06 0.1  0.4  0.09]]  max: 0.4
Working is great if it pays the bills. [[0.04 0.03 0.36 0.12 0.04 0.16 0.18 0.06]]   m
ax: 0.36
Treatment broadspectrum betalactam antibiotics including sulfonamide trimethoprim asso
ciated diagnosis. [[0.04 0.01 0.32 0.01 0.19 0.12 0.28 0.03]]       max: 0.32
```

The max value of the array is the generated prediction If all the same probabilities, the variable the bow_transformer was trained on wasn't the reviews or comments

In [ ]: