Data-X (IEOR 290): HW 8

Uvika Chaturvedi

SID: 3032162420

Homework 8: NLP & NLTK - sentiment analysis on movie reviews

Using Natural Language Processing in Python

Source: https://www.kaggle.com/c/word2vec-nlp-tutorial/data (https://www.kaggle.com/c/word2vec-nlp-tutorial/data)

Code snippets given run on both Python 2.7 and Python 3.5

Topics covered:

- Data exploration
- Text preprocessing
- Stemming
- Part of Speech (POS) tagging
- Lemmatizing
- Stopwords (abundant words)
- Bag of Words + Feature extraction
- Sentiment prediction using Random Forest

Part 0: Pre-Setup

```
In [1]: # Remove warnings
        import warnings
        warnings.filterwarnings('ignore')

        import matplotlib.pyplot as plt
        %matplotlib inline

        from __future__ import print_function, division, absolute_import #make compatible
```

Data description

You can download the data (labeledTrainData.tsv.zip) here: https://www.kaggle.com/c/word2vec-nlp-tutorial/data (https://www.kaggle.com/c/word2vec-nlp-tutorial/data), place it in your working directory & unzip the file.

Data set

The labeled data set consists of 25,000 IMDB movie reviews. The sentiment of reviews is binary, meaning an IMDB rating < 5 results in a sentiment score of 0, and a rating >=7 have a sentiment score of 1 (no reviews with score 5 or 6 are included in the analysis). No individual movie has more than 30 reviews.

File description

- **labeledTrainData** - The labeled training set. The file is tab-delimited and has a header row followed by 25,000 rows containing an id, sentiment, and text for each review.

Data columns

- **id** - Unique ID of each review
- **sentiment** - Sentiment of the review; 1 for positive reviews and 0 for negative reviews
- **review** - Text of the review

Read in the data

In [2]:
```
# Read in the data to a Pandas data frame
# Use header = 0 (first line contains col names)
# use delimiter=\t (columns are separated by tabs),
# use quoting=3 (Python will ignore doubled quotes)

import pandas as pd
train = pd.read_csv("labeledTrainData.tsv", header=0, \
                    delimiter="\t", quoting=3)
# train.shape should be (25000,3)
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25000 entries, 0 to 24999
Data columns (total 3 columns):
id           25000 non-null object
sentiment    25000 non-null int64
review       25000 non-null object
dtypes: int64(1), object(2)
memory usage: 586.0+ KB
```

Q1.1:

- 1: How many movie reviews are positive and how many are negative in labeledTrainData.tsv?

- 2: What is the average length of all the reviews (string length)?

In [3]:
```
## Input answer ##
import numpy as np

#1.
positive = train[train['sentiment'] == 1]
negative = train[train['sentiment'] == 0]
no_of_positive = len(positive)
no_of_negative = len(negative)
print ("Number of positive movie reviews is: ", no_of_positive)
print ("Number of negative movie reviews is: :", no_of_negative)

#2.
len_of_reviews = np.array([])
for val in train['review']:
    temp = len(val)
    len_of_reviews = np.append(len_of_reviews,temp)

print ("The average length of all the reviews: ",len_of_reviews.mean())
```

```
Number of positive movie reviews is:  12500
Number of negative movie reviews is: : 12500
The average length of all the reviews:  1329.95384
```

Q1:2 Explore NLP on one review

In Q1:2 you should work on the third review, i.e. `train['review'][2]`

First we would like to clean up the reviews. As you can see many interviews contain \ characters in front of quotation symobols, "`<br/>` tags, numbers, abbrevations etc.

- 1: Remove all the HTML tags in the third review, by creating a beatifulsoup object and then using the `.text` method. Save results in variable `review3`
- 2: Import NLTK's sent_tokenizer and count the number of sentences in review 3 (cleaned from HTML tags). To import sent_tokenizer use: `from nltk.tokenize import sent_tokenize`
- 3: Remove all punctuation, numbers and special characters from the third review (cleaned from HTML tags). We can do this using Regular Expression - package `re`. Save the results in variable `review3`. Code given to you as we haven't covered this in class:

  ```
  review3 = re.sub('[^a-zA-Z]',' ',review3)
  ```
- 4: Convert all the letters to lower case (save in variable `review3`). Then split the string so that every word is one element in a list (save the list in variable `review3_words`). Note: When we split the strings into words that process is called tokenization.
- 5: Use NLTK's PorterStemmer (`from nltk.stem import PorterStemmer`). Create a new Porter stemmer (`stemmer = PorterStemmer()`) and run it on every word in `review3_words`, print the results as one string (don't overwrite the `review3_words` variable from 4). What does the PorterStemmer do?
- 6: Now we want to Part Of Speech (POS)) tag the third movie review. We will use POS labeling also called grammatical tagging. To do this import `from nltk.tag import pos_tag`. When you use pos_tag on a word it returns a token-tag pair in the form of a tuple. In NLTK's Penn Treebank POS, the abbreviation (tag) for an adjective is JJ and NN for singular nouns. Count the number of singular nouns (NN) and adjectives (JJ) in `review3_words` using NLTK's pos_tag_sents. A list of the Penn Treebank pos_tag's can be found here: http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html (http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html)

- 7: An even more sophisticated operation than stemming using the PorterStemmer is called lemmatizing. Lemmatizing, in contrast to stemming, does not create non-existent words and converts words to their synonyms. In order to use lemmatizing we need to define the wordnet POS tag. A function that takes in a POS Penn Treebank tag and converts it to a wordnet tag and then lemmatizes words in a string has been given to you below. Please extend this code and print the lemmatized third movie review. Don't save the results in any variable.

- 8: Lastly we will try to remove common words that don't carry much information. These are called stopwords. In English they could for example be 'am', 'are', 'and' etc. To import NLTK's list of stopwords you need to download the stopword corpora (`import nltk` and then `nltk.download()` if you don't have it). When that is done run `from nltk.corpus import stopwords` and create a variable for English stopwords with `eng_stopwords = stopwords.words('english')`. Use the list of English stopwords to remove all the stopwords from your list of words in the third movie review, i.e. `review3_words`. Print `review3_words` without stopwords, count the number of stopwords removed and print them as well.

In [4]:
```
import bs4 as bs
import nltk
from nltk.tokenize import sent_tokenize
import re
from nltk.stem import PorterStemmer
from nltk.tag import pos_tag
from nltk.corpus import stopwords
from nltk.corpus import wordnet
from nltk.stem import WordNetLemmatizer

eng_stopwords = stopwords.words('english')
```

In [5]:
```
#1.
review3 = train['review'][2] # the review used for analysis in Q1:2
r = bs.BeautifulSoup(review3)
review3 = r.text
print (review3)
```

```
"The film starts with a manager (Nicholas Bell) giving welcome investors (Rober
t Carradine) to Primal Park . A secret project mutating a primal animal using f
ossilized DNA, like ¨Jurassik Park¨, and some scientists resurrect one of natur
e's most fearsome predators, the Sabretooth tiger or Smilodon . Scientific ambi
tion turns deadly, however, and when the high voltage fence is opened the creat
ure escape and begins savagely stalking its prey - the human visitors , tourist
s and scientific.Meanwhile some youngsters enter in the restricted area of the
 security center and are attacked by a pack of large pre-historical animals whi
ch are deadlier and bigger . In addition , a security agent (Stacy Haiduk) and
 her mate (Brian Wimmer) fight hardly against the carnivorous Smilodons. The Sa
bretooths, themselves , of course, are the real star stars and they are astound
ing terrifyingly though not convincing. The giant animals savagely are stalking
its prey and the group run afoul and fight against one nature's most fearsome p
redators. Furthermore a third Sabretooth more dangerous and slow stalks its vic
tims.The movie delivers the goods with lots of blood and gore as beheading, hai
r-raising chills,full of scares when the Sabretooths appear with mediocre speci
al effects.The story provides exciting and stirring entertainment but it result
s to be quite boring .The giant animals are majority made by computer generator
and seem totally lousy .Middling performances though the players reacting appro
priately to becoming food.Actors give vigorously physical performances dodging
 the beasts ,running,bound and leaps or dangling over walls . And it packs a ri
diculous final deadly scene. No for small kids by realistic,gory and violent at
tack scenes . Other films about Sabretooths or Smilodon are the following : ¨Sa
bretooth(2002)¨by James R Hickox with Vanessa Angel, David Keith and John Rhys
 Davies and the much better ¨10.000 BC(2006)¨ by Roland Emmerich with with Stev
en Strait, Cliff Curtis and Camilla Belle. This motion picture filled with bloo
dy moments is badly directed by George Miller and with no originality because t
akes too many elements from previous films. Miller is an Australian director us
ually working for television (Tidal wave, Journey to the center of the earth, a
nd many others) and occasionally for cinema ( The man from Snowy river, Zeus an
d Roxanne,Robinson Crusoe ). Rating : Below average, bottom of barrel."
```

In [6]:
```
#2.
sent_tokenize(review3)
print ("The number of sentences in review3 is:",len(sent_tokenize(review3)))
```

```
The number of sentences in review3 is: 13
```

In [7]:
```
#3.
review3 = re.sub('[^a-zA-Z]',' ',review3)
review3
```

Out[7]:
```
u' The film starts with a manager  Nicholas Bell  giving welcome investors  Rob
ert Carradine  to Primal Park   A secret project mutating a primal animal using
fossilized DNA  like  Jurassik Park   and some scientists resurrect one of natu
re s most fearsome predators  the Sabretooth tiger or Smilodon   Scientific amb
ition turns deadly  however  and when the high voltage fence is opened the crea
ture escape and begins savagely stalking its prey   the human visitors   touris
ts and scientific Meanwhile some youngsters enter in the restricted area of the
security center and are attacked by a pack of large pre historical animals whic
h are deadlier and bigger   In addition   a security agent  Stacy Haiduk  and h
er mate  Brian Wimmer  fight hardly against the carnivorous Smilodons  The Sabr
etooths  themselves   of course  are the real star stars and they are astoundin
g terrifyingly though not convincing  The giant animals savagely are stalking i
ts prey and the group run afoul and fight against one nature s most fearsome pr
edators  Furthermore a third Sabretooth more dangerous and slow stalks its vict
ims The movie delivers the goods with lots of blood and gore as beheading  hair
raising chills full of scares when the Sabretooths appear with mediocre special
effects The story provides exciting and stirring entertainment but it results t
o be quite boring  The giant animals are majority made by computer generator an
d seem totally lousy  Middling performances though the players reacting appropr
iately to becoming food Actors give vigorously physical performances dodging th
e beasts  running bound and leaps or dangling over walls   And it packs a ridic
ulous final deadly scene  No for small kids by realistic gory and violent attac
k scenes   Other films about Sabretooths or Smilodon are the following    Sabre
tooth       by James R Hickox with Vanessa Angel  David Keith and John Rhys Dav
ies and the much better         BC        by Roland Emmerich with with Steven S
trait  Cliff Curtis and Camilla Belle  This motion picture filled with bloody m
oments is badly directed by George Miller and with no originality because takes
too many elements from previous films  Miller is an Australian director usually
working for television  Tidal wave  Journey to the center of the earth  and man
y others  and occasionally for cinema   The man from Snowy river  Zeus and Roxa
nne Robinson Crusoe    Rating   Below average  bottom of barrel  '
```

```
In [8]: #4.
        review3 = review3.lower()
        review3_words = nltk.word_tokenize(review3)
        review3_words
```

```
Out[8]: [u'the',
         u'film',
         u'starts',
         u'with',
         u'a',
         u'manager',
         u'nicholas',
         u'bell',
         u'giving',
         u'welcome',
         u'investors',
         u'robert',
         u'carradine',
         u'to',
         u'primal',
         u'park',
         u'a',
         u'secret',
         u'project',
         u'mutating',
         u'a',
         u'primal',
         u'animal',
         u'using',
         u'fossilized',
         u'dna',
         u'like',
         u'jurassik',
         u'park',
         u'and',
         u'some',
         u'scientists',
         u'resurrect',
         u'one',
         u'of',
         u'nature',
         u's',
         u'most',
         u'fearsome',
         u'predators',
         u'the',
         u'sabretooth',
         u'tiger',
         u'or',
         u'smilodon',
         u'scientific',
         u'ambition',
         u'turns',
         u'deadly',
         u'however',
         u'and',
         u'when',
         u'the',
         u'high',
         u'voltage',
         u'fence',
         u'is',
         u'opened',
         u'the',
         u'creature',
           u'escape',
           u'and',
           u'begins',
           u'savagely',
           u'stalking',
           u'its',
           u'prey',
           u'the',
           u'human',
           u'visitors',
           u'tourists',
           u'and',
           u'scientific',
           u'meanwhile',
           u'some',
           u'youngsters',
           u'enter',
           u'in',
           u'the',
           u'restricted',
           u'area',
           u'of',
           u'the',
           u'security',
           u'center',
           u'and',
           u'are',
           u'attacked',
           u'by',
           u'a',
           u'pack',
           u'of',
           u'large',
           u'pre',
           u'historical',
           u'animals',
           u'which',
           u'are',
           u'deadlier',
           u'and',
           u'bigger',
           u'in',
           u'addition',
           u'a',
           u'security',
           u'agent',
           u'stacy',
           u'haiduk',
           u'and',
           u'her',
           u'mate',
           u'brian',
           u'wimmer',
           u'fight',
           u'hardly',
           u'against',
           u'the',
           u'carnivorous',
           u'smilodons',
           u'the',
           u'sabretooths',
           u'themselves',
           u'of',
           u'course',
           u'are',
         u'the',
         u'real',
         u'star',
         u'stars',
         u'and',
         u'they',
         u'are',
         u'astounding',
         u'terrifyingly',
         u'though',
         u'not',
         u'convincing',
         u'the',
         u'giant',
         u'animals',
         u'savagely',
         u'are',
         u'stalking',
         u'its',
         u'prey',
         u'and',
         u'the',
         u'group',
         u'run',
         u'afoul',
         u'and',
         u'fight',
         u'against',
         u'one',
         u'nature',
         u's',
         u'most',
         u'fearsome',
         u'predators',
         u'furthermore',
         u'a',
         u'third',
         u'sabretooth',
         u'more',
         u'dangerous',
         u'and',
         u'slow',
         u'stalks',
         u'its',
         u'victims',
         u'the',
         u'movie',
         u'delivers',
         u'the',
         u'goods',
         u'with',
         u'lots',
         u'of',
         u'blood',
         u'and',
         u'gore',
         u'as',
         u'beheading',
         u'hair',
         u'raising',
         u'chills',
         u'full',
         u'of',
         u'scares',
         u'when',
           u'the',
           u'sabretooths',
           u'appear',
           u'with',
           u'mediocre',
           u'special',
           u'effects',
           u'the',
           u'story',
           u'provides',
           u'exciting',
           u'and',
           u'stirring',
           u'entertainment',
           u'but',
           u'it',
           u'results',
           u'to',
           u'be',
           u'quite',
           u'boring',
           u'the',
           u'giant',
           u'animals',
           u'are',
           u'majority',
           u'made',
           u'by',
           u'computer',
           u'generator',
           u'and',
           u'seem',
           u'totally',
           u'lousy',
           u'middling',
           u'performances',
           u'though',
           u'the',
           u'players',
           u'reacting',
           u'appropriately',
           u'to',
           u'becoming',
           u'food',
           u'actors',
           u'give',
           u'vigorously',
           u'physical',
           u'performances',
           u'dodging',
           u'the',
           u'beasts',
           u'running',
           u'bound',
           u'and',
           u'leaps',
           u'or',
           u'dangling',
           u'over',
           u'walls',
           u'and',
           u'it',
           u'packs',
           u'a',
           u'ridiculous',
         u'final',
         u'deadly',
         u'scene',
         u'no',
         u'for',
         u'small',
         u'kids',
         u'by',
         u'realistic',
         u'gory',
         u'and',
         u'violent',
         u'attack',
         u'scenes',
         u'other',
         u'films',
         u'about',
         u'sabretooths',
         u'or',
         u'smilodon',
         u'are',
         u'the',
         u'following',
         u'sabretooth',
         u'by',
         u'james',
         u'r',
         u'hickox',
         u'with',
         u'vanessa',
         u'angel',
         u'david',
         u'keith',
         u'and',
         u'john',
         u'rhys',
         u'davies',
         u'and',
         u'the',
         u'much',
         u'better',
         u'bc',
         u'by',
         u'roland',
         u'emmerich',
         u'with',
         u'with',
         u'steven',
         u'strait',
         u'cliff',
         u'curtis',
         u'and',
         u'camilla',
         u'belle',
         u'this',
         u'motion',
         u'picture',
         u'filled',
         u'with',
         u'bloody',
         u'moments',
         u'is',
         u'badly',
         u'directed',
         u'by',
           u'george',
           u'miller',
           u'and',
           u'with',
           u'no',
           u'originality',
           u'because',
           u'takes',
           u'too',
           u'many',
           u'elements',
           u'from',
           u'previous',
           u'films',
           u'miller',
           u'is',
           u'an',
           u'australian',
           u'director',
           u'usually',
           u'working',
           u'for',
           u'television',
           u'tidal',
           u'wave',
           u'journey',
           u'to',
           u'the',
           u'center',
           u'of',
           u'the',
           u'earth',
           u'and',
           u'many',
           u'others',
           u'and',
           u'occasionally',
           u'for',
           u'cinema',
           u'the',
           u'man',
           u'from',
           u'snowy',
           u'river',
           u'zeus',
           u'and',
           u'roxanne',
           u'robinson',
           u'crusoe',
           u'rating',
           u'below',
           u'average',
           u'bottom',
           u'of',
           u'barrel']
```

In [9]:
```python
#5.
stemmer = PorterStemmer()
single = [stemmer.stem(words) for words in review3_words]
print(' '.join(single))

#Stemmer finds the root words. For e.g., starts is turned to start.
```

```
the film start with a manag nichola bell give welcom investor robert carradin t
o primal park a secret project mutat a primal anim use fossil dna like jurassik
park and some scientist resurrect one of natur s most fearsom predat the sabret
ooth tiger or smilodon scientif ambit turn deadli howev and when the high volta
g fenc is open the creatur escap and begin savag stalk it prey the human visito
r tourist and scientif meanwhil some youngster enter in the restrict area of th
e secur center and are attack by a pack of larg pre histor anim which are deadl
ier and bigger in addit a secur agent staci haiduk and her mate brian wimmer fi
ght hardli against the carnivor smilodon the sabretooth themselv of cours are t
he real star star and they are astound terrifyingli though not convinc the gian
t anim savag are stalk it prey and the group run afoul and fight against one na
tur s most fearsom predat furthermor a third sabretooth more danger and slow st
alk it victim the movi deliv the good with lot of blood and gore as behead hair
rais chill full of scare when the sabretooth appear with mediocr special effect
the stori provid excit and stir entertain but it result to be quit bore the gia
nt anim are major made by comput gener and seem total lousi middl perform thoug
h the player react appropri to becom food actor give vigor physic perform dodg
 the beast run bound and leap or dangl over wall and it pack a ridicul final de
adli scene no for small kid by realist gori and violent attack scene other film
about sabretooth or smilodon are the follow sabretooth by jame r hickox with va
nessa angel david keith and john rhi davi and the much better bc by roland emme
rich with with steven strait cliff curti and camilla bell thi motion pictur fil
l with bloodi moment is badli direct by georg miller and with no origin becaus
 take too mani element from previou film miller is an australian director usual
work for televis tidal wave journey to the center of the earth and mani other a
nd occasion for cinema the man from snowi river zeu and roxann robinson cruso r
ate below averag bottom of barrel
```

In [10]:
```python
#6.
l = pos_tag(review3_words)

i = 0
j = 0
for val in l:
    if val[1] == 'NN':
        i = i + 1
    elif val[1] == 'JJ':
        j = j + 1

print('The number of singular nouns: ',i)
print('The number of adjectives: ',j)
```

```
The number of singular nouns:  86
The number of adjectives:  47
```

In [11]:
```python
## 7. Example code for lemmatizing, extend it to work on the third movie review #

# example_sentence = 'gone fishing earlier than supposed to. his shirts were damag

# compare to porter stemmer (base case)
ps = PorterStemmer()
print('P. STEMMER:',' '.join([ps.stem(w) for w in review3_words]))

# Lemmatizing
wnl = WordNetLemmatizer()

def get_wordnet_pos(treebank_tag):
    '''Treebank to wordnet POS tag'''
    if treebank_tag.startswith('J'):
        return wordnet.ADJ
    elif treebank_tag.startswith('V'):
        return wordnet.VERB
    elif treebank_tag.startswith('N'):
        return wordnet.NOUN
    elif treebank_tag.startswith('R'):
        return wordnet.ADV
    else:
        return 'n' #basecase POS

# token_tag = pos_tag(example_sentence.split())

print('\nLEMMATIZER:',' '.join([wnl.lemmatize(w,pos=get_wordnet_pos(t)) for w,t ir
```

```
P. STEMMER: the film start with a manag nichola bell give welcom investor rober
t carradin to primal park a secret project mutat a primal anim use fossil dna l
ike jurassik park and some scientist resurrect one of natur s most fearsom pred
at the sabretooth tiger or smilodon scientif ambit turn deadli howev and when t
he high voltag fenc is open the creatur escap and begin savag stalk it prey the
human visitor tourist and scientif meanwhil some youngster enter in the restric
t area of the secur center and are attack by a pack of larg pre histor anim whi
ch are deadlier and bigger in addit a secur agent staci haiduk and her mate bri
an wimmer fight hardli against the carnivor smilodon the sabretooth themselv of
cours are the real star star and they are astound terrifyingli though not convi
nc the giant anim savag are stalk it prey and the group run afoul and fight aga
inst one natur s most fearsom predat furthermor a third sabretooth more danger
 and slow stalk it victim the movi deliv the good with lot of blood and gore as
behead hair rais chill full of scare when the sabretooth appear with mediocr sp
ecial effect the stori provid excit and stir entertain but it result to be quit
bore the giant anim are major made by comput gener and seem total lousi middl p
erform though the player react appropri to becom food actor give vigor physic p
erform dodg the beast run bound and leap or dangl over wall and it pack a ridic
ul final deadli scene no for small kid by realist gori and violent attack scene
other film about sabretooth or smilodon are the follow sabretooth by jame r hic
kox with vanessa angel david keith and john rhi davi and the much better bc by
 roland emmerich with with steven strait cliff curti and camilla bell thi motio
n pictur fill with bloodi moment is badli direct by georg miller and with no or
igin becaus take too mani element from previou film miller is an australian dir
ector usual work for televis tidal wave journey to the center of the earth and
 mani other and occasion for cinema the man from snowi river zeu and roxann rob
inson cruso rate below averag bottom of barrel

LEMMATIZER: the film start with a manager nicholas bell give welcome investor r
obert carradine to primal park a secret project mutate a primal animal use foss
ilized dna like jurassik park and some scientist resurrect one of nature s most
fearsome predator the sabretooth tiger or smilodon scientific ambition turn dea
dly however and when the high voltage fence be open the creature escape and beg
in savagely stalk it prey the human visitor tourist and scientific meanwhile so
me youngster enter in the restricted area of the security center and be attack
```

by a pack of large pre historical animal which be deadlier and big in addition
a security agent stacy haiduk and her mate brian wimmer fight hardly against th
e carnivorous smilodons the sabretooths themselves of course be the real star s
tar and they be astound terrifyingly though not convince the giant animal savag
ely be stalk it prey and the group run afoul and fight against one nature s mos
t fearsome predator furthermore a third sabretooth more dangerous and slow stal
k it victim the movie deliver the good with lot of blood and gore a behead hair
raise chill full of scare when the sabretooths appear with mediocre special eff
ect the story provide exciting and stirring entertainment but it result to be q
uite bore the giant animal be majority make by computer generator and seem tota
lly lousy middle performance though the player react appropriately to become fo
od actor give vigorously physical performance dodge the beast run bound and lea
p or dangle over wall and it pack a ridiculous final deadly scene no for small
kid by realistic gory and violent attack scenes other film about sabretooths o
r smilodon be the follow sabretooth by james r hickox with vanessa angel david
keith and john rhys davy and the much good bc by roland emmerich with with ste
ven strait cliff curtis and camilla belle this motion picture fill with bloody
moment be badly direct by george miller and with no originality because take t
oo many element from previous film miller be an australian director usually wor
k for television tidal wave journey to the center of the earth and many others
and occasionally for cinema the man from snowy river zeus and roxanne robinson
crusoe rating below average bottom of barrel

In [12]:
```
#8.
no_stop_words = [word for word in review3_words if word not in eng_stopwords]
no_stop_words = ' '.join([word for word in no_stop_words])
print ("review3_words without stopwords: \n\n",no_stop_words)

stop_words = [word for word in review3_words if word not in no_stop_words]
stop_words2 = ' '.join([word for word in stop_words])
print ("\n\nreview3_words's stopwords: \n\n",stop_words2)
print ("\n\nNumber of review3_words's stopwords:",len(stop_words))
```

```
review3_words without stopwords: 

 film starts manager nicholas bell giving welcome investors robert carradine pr
imal park secret project mutating primal animal using fossilized dna like juras
sik park scientists resurrect one nature fearsome predators sabretooth tiger sm
ilodon scientific ambition turns deadly however high voltage fence opened creat
ure escape begins savagely stalking prey human visitors tourists scientific mea
nwhile youngsters enter restricted area security center attacked pack large pre
historical animals deadlier bigger addition security agent stacy haiduk mate br
ian wimmer fight hardly carnivorous smilodons sabretooths course real star star
s astounding terrifyingly though convincing giant animals savagely stalking pre
y group run afoul fight one nature fearsome predators furthermore third sabreto
oth dangerous slow stalks victims movie delivers goods lots blood gore beheadin
g hair raising chills full scares sabretooths appear mediocre special effects s
tory provides exciting stirring entertainment results quite boring giant animal
s majority made computer generator seem totally lousy middling performances tho
ugh players reacting appropriately becoming food actors give vigorously physica
l performances dodging beasts running bound leaps dangling walls packs ridiculo
us final deadly scene small kids realistic gory violent attack scenes films sab
retooths smilodon following sabretooth james r hickox vanessa angel david keith
john rhys davies much better bc roland emmerich steven strait cliff curtis cami
lla belle motion picture filled bloody moments badly directed george miller ori
ginality takes many elements previous films miller australian director usually
 working television tidal wave journey center earth many others occasionally ci
nema man snowy river zeus roxanne robinson crusoe rating average bottom barrel


review3_words's stopwords: 

 with of most when its of by of which against themselves of they not its agains
t most its with of of when with but by over by about by with by with with this
 with by with because from of from below of


Number of review3_words's stopwords: 43
```

Q1:3

- 1: Create a function called `review_cleaner` that reads in a review and
  - Removes HTML tags (using beautifulsoup)
  - Removes non-letters (using regular expression)
  - Converts all words to lowercase letters and tokenizes them (using .split() method on the review strings, so that every word in the review is an element in a list)
  - Removes all the English stopwords from the list of movie review words
  - Joins the words back into one string seperated by space

NOTE: Transform the list of stopwords to a set before removing the stopwords. I.e. assign `eng_stopwords = set(stopwords.words("english"))`. Use the set to look up stopwords. This will speed up the computations A LOT (Python is much quicker when searching a set than a list).

- 2: Create three lists:

- `review_clean_original`, `review_clean_ps` and `review_clean_wnl`. Where `review_clean_original` contains all the reviews from the train DataFrame, that have been cleaned by the function `review_cleaner` defined in 1.
- `review_clean_ps` applies the PorterStemmer to the reviews in `review_clean_original`. **Note:** NLTK version 3.2.2 crashes when trying to use the PorterStemming on the string 'oed' (known bug). Therefore, use an if statement to skip just that specific string/word.
- `review_clean_wnl` contains words that have been lemmatized using NLTK's WordNetLemmatizer on the words in the list `review_clean_original`.

Note, problem 2: can take more than 10minutes to run on a laptop

In [13]:
```
## Part 1

def review_cleaner(review):
    ## EXTEND THIS FUNCTION SO THAT IT COMPLETES THE FOLLOWING STEPS: ##
    '''
    Clean and preprocess a review.

    1. Remove HTML tags
    2. Use regex to remove all special characters (only keep letters)
    3. Make strings to lower case and tokenize / word split reviews
    4. Remove English stopwords
    5. Rejoin to one string
    '''

    #1.
    r = bs.BeautifulSoup(review)
    review = r.text

    #2.
    review = re.sub('[^a-zA-Z]',' ',review)

    #3.
    review = review.lower()
    review = nltk.word_tokenize(review)

    #4. CHECK FOR STOPWORDS IN: eng_stopwords
    eng_stopwords = set(stopwords.words("english"))
    review = [w for w in review if not w in eng_stopwords]

    #5.
    review = ' '.join([word for word in review])
    #6.
    return(review)
```

In [14]:
```
## Part 2

## Step 1: Clean up all the original reviews
num_reviews = len(train['review'])

review_clean_original = []

for i in range(0,num_reviews):
    if( (i+1)%500 == 0 ):
        # print progress
        print("Done with %d reviews for review_clean_original" %(i+1))
    review_clean_original.append(review_cleaner(train['review'][i]))
```

```
Done with 500 reviews for review_clean_original
Done with 1000 reviews for review_clean_original
Done with 1500 reviews for review_clean_original
Done with 2000 reviews for review_clean_original
Done with 2500 reviews for review_clean_original
Done with 3000 reviews for review_clean_original
Done with 3500 reviews for review_clean_original
Done with 4000 reviews for review_clean_original
Done with 4500 reviews for review_clean_original
Done with 5000 reviews for review_clean_original
Done with 5500 reviews for review_clean_original
Done with 6000 reviews for review_clean_original
Done with 6500 reviews for review_clean_original
Done with 7000 reviews for review_clean_original
Done with 7500 reviews for review_clean_original
Done with 8000 reviews for review_clean_original
Done with 8500 reviews for review_clean_original
Done with 9000 reviews for review_clean_original
Done with 9500 reviews for review_clean_original
Done with 10000 reviews for review_clean_original
Done with 10500 reviews for review_clean_original
Done with 11000 reviews for review_clean_original
Done with 11500 reviews for review_clean_original
Done with 12000 reviews for review_clean_original
Done with 12500 reviews for review_clean_original
Done with 13000 reviews for review_clean_original
Done with 13500 reviews for review_clean_original
Done with 14000 reviews for review_clean_original
Done with 14500 reviews for review_clean_original
Done with 15000 reviews for review_clean_original
Done with 15500 reviews for review_clean_original
Done with 16000 reviews for review_clean_original
Done with 16500 reviews for review_clean_original
Done with 17000 reviews for review_clean_original
Done with 17500 reviews for review_clean_original
Done with 18000 reviews for review_clean_original
Done with 18500 reviews for review_clean_original
Done with 19000 reviews for review_clean_original
Done with 19500 reviews for review_clean_original
Done with 20000 reviews for review_clean_original
Done with 20500 reviews for review_clean_original
Done with 21000 reviews for review_clean_original
Done with 21500 reviews for review_clean_original
Done with 22000 reviews for review_clean_original
Done with 22500 reviews for review_clean_original
Done with 23000 reviews for review_clean_original
Done with 23500 reviews for review_clean_original
Done with 24000 reviews for review_clean_original
Done with 24500 reviews for review_clean_original
Done with 25000 reviews for review_clean_original
```

```
In [19]: review_clean_original[0]
```

```
Out[19]: u'stuff going moment mj started listening music watching odd documentary watche
         d wiz watched moonwalker maybe want get certain insight guy thought really cool
         eighties maybe make mind whether guilty innocent moonwalker part biography part
         feature film remember going see cinema originally released subtle messages mj f
         eeling towards press also obvious message drugs bad kay visually impressive cou
         rse michael jackson unless remotely like mj anyway going hate find boring may c
         all mj egotist consenting making movie mj fans would say made fans true really
          nice actual feature film bit finally starts minutes excluding smooth criminal
          sequence joe pesci convincing psychopathic powerful drug lord wants mj dead ba
         d beyond mj overheard plans nah joe pesci character ranted wanted people know s
         upplying drugs etc dunno maybe hates mj music lots cool things like mj turning
          car robot whole speed demon sequence also director must patience saint came fi
         lming kiddy bad sequence usually directors hate working one kid let alone whole
         bunch performing complex dance scene bottom line movie people like mj one level
         another think people stay away try give wholesome message ironically mj bestest
         buddy movie girl michael jackson truly one talented people ever grace planet gu
         ilty well attention gave subject hmmm well know people different behind closed
          doors know fact either extremely nice stupid guy one sickest liars hope latte
         r'
```

```
In [17]: ### Step2: Use review_clean_original to create review_clean_ps using PorterStemmer
         ps = PorterStemmer()
         review_clean_ps = []
         for val in review_clean_original:
             temp = ' '.join([ps.stem(w) for w in val.split() if 'oed' not in w])
             review_clean_ps.append(temp)
```

```
In [18]: review_clean_ps[0]
```

```
Out[18]: u'stuff go moment mj start listen music watch odd documentari watch wiz watch m
         oonwalk mayb want get certain insight guy thought realli cool eighti mayb make
          mind whether guilti innoc moonwalk part biographi part featur film rememb go s
         ee cinema origin releas subtl messag mj feel toward press also obviou messag dr
         ug bad kay visual impress cours michael jackson unless remot like mj anyway go
          hate find bore may call mj egotist consent make movi mj fan would say made fan
         true realli nice actual featur film bit final start minut exclud smooth crimin
          sequenc joe pesci convinc psychopath power drug lord want mj dead bad beyond m
         j overheard plan nah joe pesci charact rant want peopl know suppli drug etc dun
         no mayb hate mj music lot cool thing like mj turn car robot whole speed demon s
         equenc also director must patienc saint came film kiddi bad sequenc usual direc
         tor hate work one kid let alon whole bunch perform complex danc scene bottom li
         ne movi peopl like mj one level anoth think peopl stay away tri give wholesom m
         essag iron mj bestest buddi movi girl michael jackson truli one talent peopl ev
         er grace planet guilti well attent gave subject hmmm well know peopl differ beh
         ind close door know fact either extrem nice stupid guy one sickest liar hope la
         tter'
```

In [20]:
```
### Step3: Use review_clean original to create review_clean_wnl using the WordNetL
wnl = WordNetLemmatizer()

def get_wordnet_pos(treebank_tag):
    '''Treebank to wordnet POS tag'''
    if treebank_tag.startswith('J'):
        return wordnet.ADJ
    elif treebank_tag.startswith('V'):
        return wordnet.VERB
    elif treebank_tag.startswith('N'):
        return wordnet.NOUN
    elif treebank_tag.startswith('R'):
        return wordnet.ADV
    else:
        return 'n' #basecase POS

# token_tag = pos_tag(example_sentence.split())

review_clean_wnl = []
for val in review_clean_original:
    l = pos_tag(val.split())
    temp = ' '.join([wnl.lemmatize(w,pos=get_wordnet_pos(t)) for w,t in l])
    review_clean_wnl.append(temp)
```

In [21]:
```
review_clean_wnl[0]
```

Out[21]:
```
u'stuff go moment mj start listen music watch odd documentary watch wiz watch m
oonwalker maybe want get certain insight guy think really cool eighty maybe mak
e mind whether guilty innocent moonwalker part biography part feature film reme
mber go see cinema originally release subtle message mj feel towards press also
obvious message drug bad kay visually impressive course michael jackson unless
 remotely like mj anyway go hate find boring may call mj egotist consent make m
ovie mj fan would say make fan true really nice actual feature film bit finally
start minute exclude smooth criminal sequence joe pesci convince psychopathic p
owerful drug lord want mj dead bad beyond mj overheard plan nah joe pesci chara
cter rant wanted people know supply drug etc dunno maybe hat mj music lot cool
 thing like mj turn car robot whole speed demon sequence also director must pat
ience saint come film kiddy bad sequence usually director hate work one kid let
alone whole bunch perform complex dance scene bottom line movie people like mj
 one level another think people stay away try give wholesome message ironically
mj best buddy movie girl michael jackson truly one talented people ever grace p
lanet guilty well attention give subject hmmm well know people different behind
closed door know fact either extremely nice stupid guy one sickest liars hope l
atter'
```

Q1:4: Feature vectors and Bag of Words model

Explanation

Derived from source: https://www.kaggle.com/c/word2vec-nlp-tutorial/details/part-1-for-beginners-bag-of-words (https://www.kaggle.com/c/word2vec-nlp-tutorial/details/part-1-for-beginners-bag-of-words)

We will now use scikit-learn to create numeric representations of the words in the reviews, using a method called Bag of Words. You can see this as learning a vocabulary from all the reviews and counting how many times a word appears in the reviews. For example, if we have two sentences:

Sentence 1: "cool students study cool data science"

Sentence 2: "to know data science study data science"

The vocabulary of these two sentences can be summarized in a dictionary:

{ cool, students, study, data, science, to, know }

The bags of words count the number of times each word occur in a sentence. In Sentence 1, "cool" appears twice, and "students", "study", "data", and "science" appear once. The feature vector for Sentence 1 is:

Sentence 1: { 2, 1, 1, 1, 1, 0, 0 }

And for Sentence 2: { 0, 0, 1, 2, 2, 1, 1}

Applying this strategy to the IMDB movie reviews

The movie review data contains a lot of words. To limit the analysis we use the 5000 most frequent words from the cleaned reviews. To extract the bag of words features we will use scitkit-learn.

The training data will be created by the `CountVectorizer` function from scikit-learn, and the training array will have 25000 rows (one for each review) and 5000 features (one for each vocabulary word).

CountVectorizer can automatically handle text cleaning, but here we specify "None", instead we did a step-by-step cleaning of the data in the earlier problems.

Random Forest for review sentiment classification

First split up the data set so that 80% are used as training samples (the first 20000 reviews and their sentiment) and 20% are used as validation samples (the last 5000 reviews and their sentiment). Use Random Forest to do numeric training on the features for the training samples from the Bag of Words and their respective sentiment labels for each review / feature vector. The number of trees is set to 50 as a default value.

Problem

- 1: Run this analysis for the three cleaned review lists, i.e. `review_clean_original`, `review_clean_ps` and `review_clean_wnl`, by using the code below. Extend the function to print the **validation accuracy** by using `forest.predict(train_data_features[20000:,:])` and then comparing the resulting sentiment predictions with the ones stored in `train["sentiment"][20000:]`. **Note:** The printed validation accuracy should show the percentage of correctly predicted sentiments for the validation set.

- 2: Print the validation accuracy obtained for the three models. **Note:** Takes about 4mins to run.

- 3: What data preprocessing strategy worked the best? Why do you think that is? (Feel free to change the number of features extracted in the bag of words model and the number of trees in the random forest model, to see how it effects your accuracy).

In [27]:
```python
# %%time # times the operation

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import RandomForestClassifier
import numpy as np
from sklearn.metrics import accuracy_score

def predict_sentiment(X,y=train["sentiment"]):


    print("Creating the bag of words model!\n")
    # CountVectorizer" is scikit-learn's bag of words tool.
    vectorizer = CountVectorizer(analyzer = "word",   \
                                 tokenizer = None,    \
                                 preprocessor = None, \
                                 stop_words = None,   \
                                 max_features = 5000)

    # Then we use fit_transform() to fit the model / learn the vocabulary,
    # then transform the data into feature vectors.
    # The input should be a list of strings. .toarraty() converts to a numpy arra
    train_data_features = vectorizer.fit_transform(X).toarray()

    # You can extract the vocabulary created by CountVectorizer
    # by running print(vectorizer.get_feature_names())


    print("Training the random forest classifier!\n")
    # Initialize a Random Forest classifier with 50 trees
    forest = RandomForestClassifier(n_estimators = 50)

    # Fit the forest to the training set, using the bag of words as
    # features and the sentiment labels as the target variable
    forest = forest.fit(train_data_features[0:20000,:], y[0:20000] )


    '''## MAKE PREDICTIONS HERE ##'''
    prediction = forest.predict(train_data_features[20000:,:])
    accuracy  = accuracy_score(train["sentiment"][20000:],prediction)
    return accuracy

# '''Then run'''
original = predict_sentiment(review_clean_original)
print ("Accuracy for review_clean_original:",original)

ps = predict_sentiment(review_clean_ps)
print ("Accuracy for review_clean_ps:",ps)

wnl = predict_sentiment(review_clean_wnl)
print ("Accuracy for review_clean_wnl:",wnl)
```

```
Creating the bag of words model!

Training the random forest classifier!

Accuracy for review_clean_original: 0.8326
Creating the bag of words model!

Training the random forest classifier!

Accuracy for review_clean_ps: 0.8278
Creating the bag of words model!

Training the random forest classifier!
```

```
Accuracy for review_clean_wnl: 0.8296
```

In [ ]:
```
#According to the accuracy, it seems like the review_clean_original had the maximu
#not stem the words.
```

Extra Credit (worth 1p)

- **Question:** Preprocess the reviews in any way you find suitable and build your own ML model that can predict the sentiment of movie reviews. Credit will be given if you can obtain a prediction accuracy of over 90%, when predicting the sentiments of the validation set (the last 5000 reviews). Train your model on the first 20000 reviews (with their sentiment as the target variable).

In [ ]:
```
## Input Answer ##
```