Homework 2

Jiarong Ye

September 17, 2018

```
In [2]: import pandas as pd
        import datascience as ds
        from datascience import *
        import numpy as np
        import re
        import string
```

Read the tagged tweets

Sentiment

- Irrelevant: -1
- Neutral: 0
- Positive: 1
- Negative: 2

Tagging Guideline

- Irrelevant: does not include any content related to the president
  - eg:
    - `RT @SassyT_Joy: @BCMsolt @fightinirish109 @dybarb @Wolfie_V3 @KevinHu63682270 @CNS15141 @DogsBCool @Punkin682 @hirschA8720 @MrsGoodoz @MAGA`
- Neutral: does not contain any obvious emotional markers towards the president
  - eg:
    - `2 lives were just lost in #Tennessee. #POTUS its time to do something. #BackfireTrump https://t.co/5AYAqUV2at`
- Positive: contain obvious positive emotional markers towards the president
  - eg:
    - `RT @michaeljohns: .@POTUS' exceptional speech in #Shanksville: At this memorial on this sacred earth in the field beyond this wall and`
- Negative: contain obvious negative emotional markers towards the president
  - eg:

* RT @ReclaimingUSA20: Last week he couldnt even say anonymously correctly during one of his hate rallies. Today @realDonaldTrump warned

In [20]:
```
tagged= pd.read_csv('tagged.csv', sep= ',')
tagged['sentiment'] = tagged['sentiment'].astype(dtype=int)
```

Clean the tweet text and remove nan

In [21]:
```
def process_text(data):
    cleaned_text = [
        re.sub('\s+', ' ',
    re.sub("([^0-9A-Za-z \t])|(\w+:\/\/\S+)|^rt|http.+?", '',
    tweets.lower()).strip(string.punctuation).strip()) for tweets in data
    ]
    return cleaned_text

tagged['text'] = process_text(tagged['text'])
tagged = tagged.drop(tagged.columns[0], axis=1).reset_index(drop=True)
```

In [22]:
```
not_null_text = 1 ^ pd.isnull(tagged["text"])
not_null_sentiment = 1 ^ pd.isnull(tagged["sentiment"])
tagged = tagged.loc[not_null_sentiment & not_null_text, :]
```

In [23]:
```
len(tagged)
```

Out[23]: 4817

Remove duplicates

In [24]:
```
tagged = tagged.iloc[tagged['text'].drop_duplicates().index]
tagged = tagged.reset_index(drop=True)
tagged = tagged.dropna(axis=0)
tagged = tagged.reset_index(drop=True)
tagged.to_csv('cleaned_tagged.csv', sep=',')
tagged.head()
```

Out[24]:

	user_id	user_name	\
0	802657195661742080	Christine Warren	
1	1039245812230893570	Trumpservative	
2	282084840	Darrel Sheldon #MAGAVETERAN	
3	62315639	Queer Liberal Voting Snowflake	
4	823307049266245633	don jones #veteran (K)	

	tweet_time	location	\
0	Wed Sep 12 01:38:14 +0000 2018	Fremont CA	
1	Wed Sep 12 01:38:16 +0000 2018	Tulsa OK	
2	Wed Sep 12 01:38:18 +0000 2018	None	
3	Wed Sep 12 01:38:18 +0000 2018	Big Sandy Texas	

```
4  Wed Sep 12 01:38:19 +0000 2018    United States

                                                text  sentiment
0  2 american lives were just lost in tennessee p...          0
1  realfarmacist realdonaldtrump tuckercarlson ou...          1
2  rightgottweets rev out the trumptrain for blue...          0
3  tennessee is suffering after shooting takes 2 ...          0
4  sassytjoy bcmsolt fightinirish109 dybarb wolfi...         -1
```

Count the tweets

```
In [25]: tagged = ds.Table.read_table('cleaned_tagged.csv', sep=',')

In [26]: def count_tweets(table):
             classes = {
                        'irrelevant':-1,
                        'neutral':0,
                        'positive':1,
                        'negative':2
                       }
             for i in classes:
                 c =  table.where('sentiment', are.equal_to(classes[i])).num_rows
                 print('There are {} {} tweets in the data'.format(c, i))

In [27]: count_tweets(tagged)

There are 163 irrelevant tweets in the data
There are 470 neutral tweets in the data
There are 415 positive tweets in the data
There are 248 negative tweets in the data
```