CMPS242 HW5 Report

Team Member: Yunzhe Li(#1571061) & Yong Deng(#1571065)

Section I: word2vec

- In this section, we import the dataset 'train.csv' 'test.csv' and convert them into numerical vectors using library Spacy's wording embedding.
- According to the official documentation of Spacy, its pre-trained build-in dictionary is actually from the GloVe with 300 dimensions per word vector, which I believe will greatly improve our accuracy.

In [1]:

```
## Import all the libraries I'll use in this section.
## Here the 'en_vectors_web_lg' is the dictionary we will use.

import pandas as pd
import en_vectors_web_lg
import numpy as np
```

Read the dataset

In [2]:

```
## Define a funtion to read the .csv file and split the labels and tweets into tw
o list.
## Note that the labels of test.csv is None, thus we discard them directly.

def csv_reading(f_name):

    with open(f_name, "r", encoding='utf8') as train:
        csvfile = pd.read_csv(train)

    csvfile = csvfile.values.tolist()

    labels = [row[0] for row in csvfile]
    twitters = [row[1] for row in csvfile]

    return labels, twitters
```

In [3]:

```
## Run the .csv file reading function

train_labels, train_twitters = csv_reading('train.csv')
_, test_twitters = csv_reading('test.csv')
```

Remove the urls

In [4]:

```
## Define a function to remove all the urls at the end of each twitter.

def remove_url(twitters):

    twitters_iter = twitters.__iter__()

    for i in range(len(twitters)):
        twitters[i] = twitters_iter.__next__().split('http')[0]

    return twitters
```

In [5]:

```
train_twitters = remove_url(train_twitters)
test_twitters = remove_url(test_twitters)
```

Apply the Spacy dictionary to implement words embedding

In [6]:

```
## load the dictionary

nlp = en_vectors_web_lg.load()
```

In [7]:

```
## Define the word2vec function.
## It will return a list of numpy ndarrays which has different length.
## Each tweet will convert to a numpy array with shape [length, 300].

def word2vec(twitters):

    twitters_vectors = [None]*len(twitters)

    for i in range(len(twitters)):

        twitter_doc = nlp(twitters[i])
        twitter_vector = [None]*len(twitter_doc)

        for j in range(len(twitter_doc)):
            twitter_vector[j] = twitter_doc[j].vector

        twitters_vectors[i] = twitter_vector

    return twitters_vectors
```

In [8]:

```
## Run the word2vec function

train_twitters = word2vec(train_twitters)
test_twitters = word2vec(test_twitters)
```

Convert the binary cases labels into 1 and 0

In [9]:

```
## Convert the label 'HillaryClinton' to 0 and 'realDonaldTrump' to 1

def numeric_label(labels):
    for i in range(len(labels)):
        if labels[i] == 'HillaryClinton':
            labels[i] = 0
        elif labels[i] == 'realDonaldTrump':
            labels[i] = 1

    return labels
```

In [10]:

```
train_labels = numeric_label(train_labels)
```

Save my embedding results into a .npz file for Section II use

In [11]:

```
np.savez('embedding_matrix.npz', train_matrix=train_twitters, test_matrix=test_tw
itters, train_labels=train_labels)
```