Twitter Data - Prepare Corpus

July 19, 2019

- Shubhendu Vimal – 11915067
- Anmol More - 11915043

References : - https://towardsdatascience.com/extracting-twitter-data-pre-processing-and-sentiment-analysis-using-python-3-0-7192bd8b47cf - https://towardsdatascience.com/with-the-emergence-of-social-media-high-quality-of-structured-and-unstructured-information-shared-b16103f8bb2e - https://pypi.org/project/tweet-preprocessor/ - https://towardsdatascience.com/another-twitter-sentiment-analysis-bb5b01ebad90

In [ ]:
```
#import necessary libraries
import pandas as pd
from textblob import TextBlob
import matplotlib.pyplot as plt
import os
import re
from collections import Counter
import glob

from wordcloud import WordCloud

from bs4 import BeautifulSoup
from nltk.corpus import stopwords
import nltk
from nltk.stem import PorterStemmer
from nltk import word_tokenize

from sklearn.feature_extraction.text import CountVectorizer
```

Read Data
read zomato and swiggy tweets from csv folder (all 45 days of tweets)

In [ ]:
```
swiggy = pd.DataFrame()
for file_name in glob.glob("data/swiggy/"+'*.csv'):
    df = pd.read_csv(file_name)
    swiggy = swiggy.append(df, sort=False)
swiggy['length'] = swiggy['full_text'].apply(len)
```

```
In [ ]: zomato = pd.DataFrame()
        for file_name in glob.glob("data/zomato/"+'*.csv'):
            df = pd.read_csv(file_name)
            zomato = zomato.append(df, sort=False)
        zomato['length'] = zomato['full_text'].apply(len)

In [ ]: #remove retweets, same tweets, tweets by self
        print(swiggy.shape)
        swiggy = swiggy[~swiggy['full_text'].str.startswith("RT")]
        swiggy = swiggy.drop_duplicates(subset=['full_text'], keep="first")
        swiggy = swiggy[~((swiggy['screen_name'] == 'swiggy_in') | (swiggy['screen_name'] == '
        print(swiggy.shape)

In [ ]: swiggy.sample(5)

In [ ]: #remove retweets, same tweets, tweets by self
        print(zomato.shape)
        zomato = zomato[~zomato['full_text'].str.startswith("RT")]
        zomato = zomato.drop_duplicates(subset=['date','full_text'], keep="first")
        zomato = zomato[~((zomato['screen_name'] == 'zomatoIn') | (zomato['screen_name'] == 'z
        print(zomato.shape)
        zomato.sample(5)

In [ ]: #save raw data before doing further analysis
        zomato.to_csv('data/zomato_raw.csv')
        swiggy.to_csv('data/swiggy_raw.csv')
```

Check lengths of swiggy and zomato text

```
In [ ]: fig, ax = plt.subplots(figsize=(5, 5))
        plt.boxplot(swiggy['length'])
        plt.show()

In [ ]: fig, ax = plt.subplots(figsize=(5, 5))
        plt.boxplot(zomato['length'])
        plt.show()

In [ ]: #Check sample tweets
        swiggy['full_text'].sample(20)
```

Clean data

```
In [ ]: import preprocessor as p

        stop_words = set(stopwords.words('english'))
        stop_words_list = list(stop_words)

        extended_list = []
        with open('stop_word_extended.txt') as f:
```

```
        extended_list.extend([word for line in f for word in line.split()])

    stop_words_list.extend(extended_list)
    # stop_words_list.extend(stop_words_list.extend(['humans', 'water', 'may', 'nice', 'zo
    #                                                'swiggy', 'order', 'food','delivery',
    #                                                'guy', 'time', 'says', 'days', 'shall
    # stop_words_list = list(set(stop_words_list))

    #remove stop words
    def remove_stop_words(text) :
        word_tokens = word_tokenize(text)
        filtered_tokens = [w for w in word_tokens if not w in stop_words_list]
        return ' '.join(filtered_tokens)

    #clean tweets for punctuations, numbers, # etc
    def clean_tweets(text) :
        print(text)
        text = BeautifulSoup(text, 'lxml').get_text()
        try:
            text = text.decode("utf-8-sig").replace(u"\ufffd", "?")
        except:
            text = text

        text = re.sub(r'@[A-Za-z0-9_]+','',text)  #remove all @mention
        text = re.sub('https?://[A-Za-z0-9./]+','', text)  #remove links
        text = re.sub("[^a-zA-Z\s]", "", text)  #remove all #, numbers, etc non alphabets
        text = text.lower().strip() #lowercase and strip
        text = re.sub(' +', ' ', text) #all double spaces with single

        text = text.replace('delivered', 'delivery')
        text = text.replace('deliver', 'delivery')
        text = text.replace('deliveryy', 'delivery')
        text = text.replace('customers', 'customer')
        text = text.replace('guys', 'guy')
        text = text.replace('boy', 'guy')
        text = text.replace('restaurants', 'restaurant')

        text = remove_stop_words(text)
        #text = p.clean(text)
        print(text + "\n\n")
        return text
```

In [ ]:
```
#print original text and clean text
swiggy['clean_text'] = swiggy['full_text'].apply(lambda x: clean_tweets(x))
```

In [ ]:
```
#print original text and clean text
zomato['clean_text'] = zomato['full_text'].apply(lambda x: clean_tweets(x))
```

In [ ]:
```
#remove empty tweets after cleaning
```

```
zomato = zomato[zomato['clean_text'] != ""]
swiggy = swiggy[swiggy['clean_text'] != ""]
print(zomato.shape)
print(swiggy.shape)
```

```
In [ ]: zomato.to_csv('data/zomato.csv')
        swiggy.to_csv('data/swiggy.csv')
```

Final Data

```
In [ ]: zomato_text = '\n'.join(zomato['clean_text'])
        zomato_text = zomato_text + '\n'
        text_file = open("data/zomato.txt", "w")
        text_file.write(zomato_text)
        text_file.close()

        swiggy_text = '\n'.join(swiggy['clean_text'])
        swiggy_text = swiggy_text + '\n'
        text_file = open("data/swiggy.txt", "w")
        text_file.write(swiggy_text)
        text_file.close()
```

```
In [ ]: swiggy_corpus = ''.join(swiggy['clean_text'])
        zomato_corpus = ''.join(zomato['clean_text'])
```

Initial Analysis

```
In [ ]: swiggy_word_tokens = word_tokenize(swiggy_corpus)
        swiggy_corpus = ' '.join(swiggy_word_tokens)
        cloud = WordCloud(background_color="white").generate(swiggy_corpus)

        plt.figure(figsize=(15,15))
        plt.imshow(cloud)

        plt.axis('off')
        plt.show()
```

```
In [ ]: zomato_word_tokens = word_tokenize(zomato_corpus)
        zomato_corpus = ' '.join(zomato_word_tokens)
        cloud = WordCloud(background_color="white").generate(zomato_corpus)

        plt.figure(figsize=(15,15))
        plt.imshow(cloud)

        plt.axis('off')
        plt.show()
```

```
In [ ]: plt.figure(figsize=(12,5))
        plt.xticks(fontsize=13, rotation=90)
        fd = nltk.FreqDist(zomato_word_tokens)
        fd.plot(25,cumulative=False, title='Top Words for Zomato')
```

```
In [ ]: plt.figure(figsize=(12,5))
        plt.xticks(fontsize=13, rotation=90)
        fd = nltk.FreqDist(swiggy_word_tokens)
        fd.plot(25,cumulative=False, title='Top Words for Swiggy')

In [ ]:
```