基于 Twitter 平台下的简单情感分析

但孝磊 danxiaolei2008@gmail.com

微博客（Microblog），是一个基于在线网络用户关系的信息分享、传播及获取平台。作为新兴的社交媒体，微博客凭借自身的简洁性、共享性、实时性、互动性等特点，深刻的影响着人们的现今人们的生活，极大地提升了网络媒体的服务效率，其价值得到广泛认可。这种互动平台的高度自由性和参与性，使得越来越多的人习惯在其上表达自己的观点和情感。在 2007 年诞生于美国的 Twitter 截止 2013 年 10 月，已有每月 2.15 亿的用户活跃量[1]。而国内的后起之秀新浪微博，也在 2013 年 12 月达到 1.291 亿[2]。如此庞大的用户群体在上面发布、分享信息。于是，也有越来越多的人开始关注到微博客潜在的价值[3] [4] [5] [6]。其中，情感分析就是其中非常重要的一个研究方向。

情感分析主要是进行感情极性的判定，即判断一条微博客消息所表达的情感的正、负、中性。通过情感分析，我们可以知道说话者在传单信息时所隐含的情绪状态，从而对说话者的内心态度做出一定的评估。情感分析在微博客网络上地应用，将有助于了解大多微博客用户的所生存的社会 舆情走势，也可以帮助企业了解市场上对于某种产品的喜好与厌恶，亦能进行公共领域突发事件侦测，同时其在心理学、社会学等领域的研究中也有着不容小嘘的作用[7] [8]。

在这个项目中，我自己采集了一段时间的微博客数据，并对其进行简单的情感分类。

1 数据采集和资源选取

在采集数据时，考虑到 Twitter 数据采集的方便性，以及大多数 Twitter 用户的使用时间，我采集 的数据时间是 2014 年 6 月 30 日上午 9 点 15 分到 10 点 15 分（UTC+8）。虽然从 Twitter 获取的数据的结构化的 JSON 数据，然而在进行情感分析是，我们需要对每条微博客的文本信息进行情感评分。这里我使用的 AFINN 词库的 AFINN-111 版本[9]。这是由 Finn Årup Nielsen 从 2009 年到 2011 年手动标注的词典。AFINN 词库一个将英语单词按照从消极到积极的 11 个不同程度划分成从-5 到 5 的 11 个分数。AFINN-111 版本共包含常用单词短语 2477 个。我决定用这个词典，不仅因为这个词典有着出色的表现，更是因为这个词包含了单词短语的人称时态变化，使得我不用额外地再去进行分词等操作。

2 数据预处理

由于 AFINN-111 词库只对 2477 个具有情绪倾向的常用英文单词做出了评分，却不包含 Unicode（如 u’alpha’， u’2022’ 等）解码和表情符号（如☺， ☹， :-(， :)等 ），这里我主要针对英文的文本进行操作，非英文的样本将被剔除。尽管这可能会在一定程度上影响样本的抽样效果，但是当我拥有了处理 Unicode 和表情符号的词库时，这将很容易得到扩展。

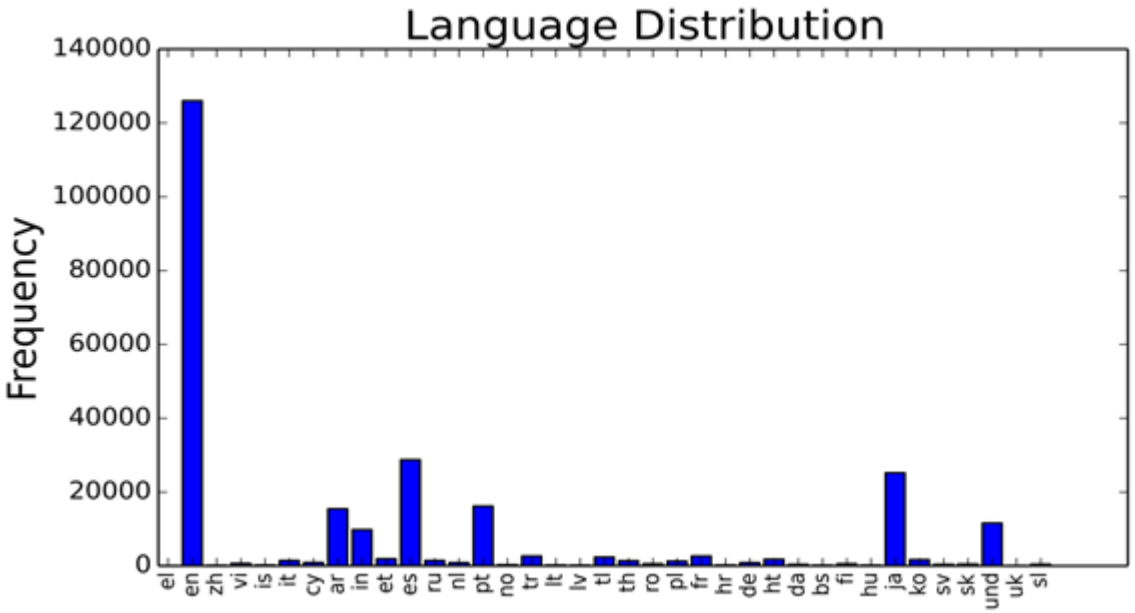

Figure 1 Language Distribution

另外，在很多的微博内容中，存在有点对点式交流，如

"CAN NOT WAIT FOR @Free_Blues THIS WEDNESDAY"

这其中的*@Free_Blues* 另外一个用户的的用户名，而其所包含的"Free"这个单词是积极的、正性的，在 Afinn 的评分中为+1，但是它却并不代表该文本的感情色彩。但是，我们不妨可以假设，该发布者的这位朋友显然不是郁郁寡欢的一个人——起码他/她用了这样一个积极的用户名。那这位发布者多多少少会受到这位积极向上的朋友的影响，即所谓的"正能量"影响，那么将这个+1 分赋予这位发布者也不是未尝不可的。因此，我在这里并没有对这种类似的非文意词语排除在外。

3 样本分析

首先，我们不妨看一下在这个数据集中，语言的分布情况（Figure 1）。非常明显地，我们可以看到在这一段时间里，Twitter 用户使用语言比较多的是英语（en）、日语（ja）和西班牙语（es）。我想这和我在采集这个数据的时间有很大关系。当然，我认为这也能在一定程度上表现出 Twitter 在日语和西班牙语的使用人群中的都有着一定的受欢迎程度，尽管我们在接下来的探究中并不包含任何非英语用户。

由于我并不在这里解决 Unicode 的编码问题，接下来的数据分析将只针对使用英语的 Twitter 条目。

在除去非英语用户群之后，我们查看了一下 Twitter 记录下来的用户登陆途径（Figure 2）。由于 Twitter 用户的登录渠道非常广泛，我在这里剔除了访问次数小于 50 次的条目。不难看出，使用 iPhone 登陆的人数遥遥领先，远远超出了排名第二的 Twitter for Android（前者大约是后者的 2.5 倍）。另外，Twitter 用户比较倾向于使用移动设备登陆。即便使用 PC，也是使用类似于

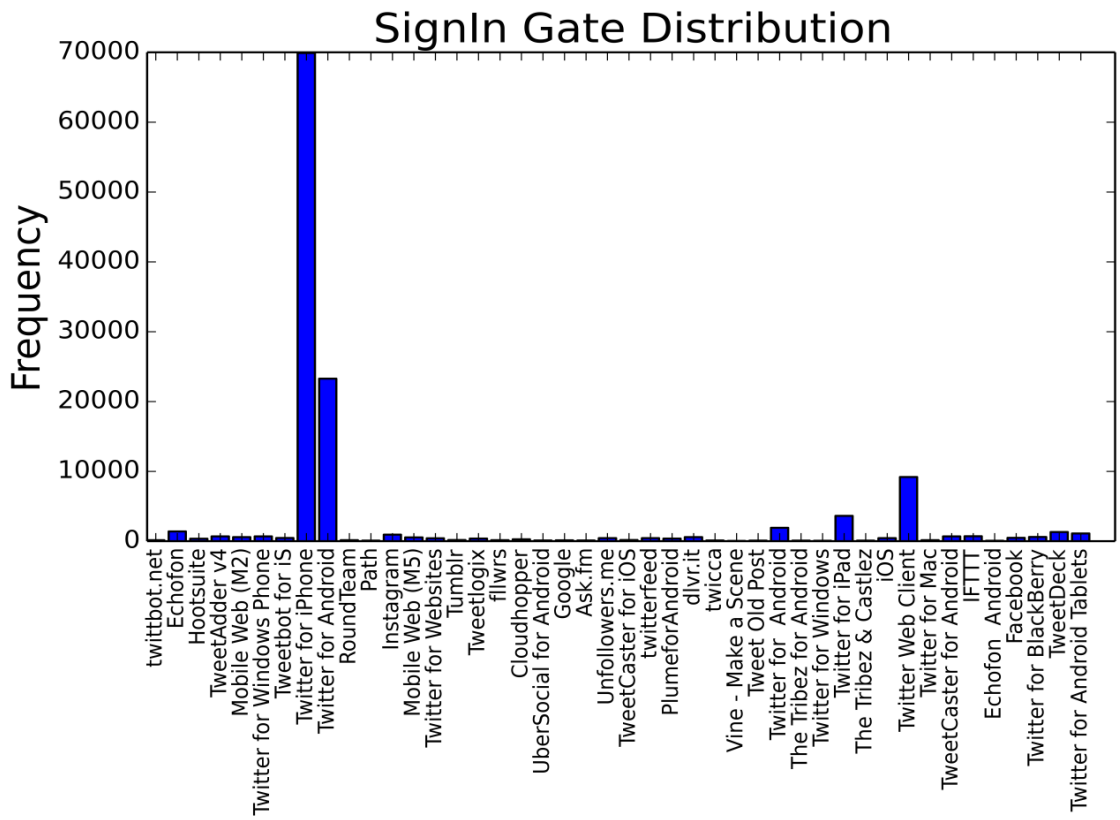

Figure 2 Sign In Source Distribution for English Users

客户端（如 Twitter Web client 等）的应用来管理 Twitter。也有一些人会使用其他的专门用来管理 Twitter 的应用来提高管理的效率（如 Echofon、tweetdeck 等）。另外还有一些用户会使用一些其他的社交网络（如 Instagram、Facebook 等）来同步管理自己的 Twitter 账户。在有关用户登陆方式的描述中，我发现有相当一部分人使用一个名叫“Twitter for Android”的入口，而不是典型的“Twitter for Android”（两者在“Android”前面相差了一个空格）。这是一个很奇怪的现象，值得关注。

那么，对于这段时间里面所产生的 Twitter 所表现的情感如何呢？通过使用 AFINN-111 词典，所有的经过前面一系列处理之后的 Twitter 文本都会被赋予一个分数。这一过程是通过在 AFINN-111 中查找一条 Twitter 文本中所有单词实现的：如果单词出现在 AFINN-111 中，则将其分数记下来，如果在 AFINN-111 没有找到某个单词，则记为 0 分，这样将一条 Twitter 文本中的所有单词的分数加起来，记为这条 Twitter 的情感分数，以此来判定其情感倾向。这样，我们便将所有的 Twitter 文本转换成一个数值化的向量了。通过所有 Twitter 文本的分数，我们可以发现，这个期间所产生的 Twitter 比较多的是徘徊在中性（0 分）左右的（Figure 3）。对这样一个分数向量进行原假设为均值等于零的 t-test，得到的 p 值近似为 0.000，且均值为 0.3944。也就是说，平均而言，情感是偏向于积极性的，正向的——至少情感分数在统计学上来说，是显著为正的，尽管“积极”的程度并不大。

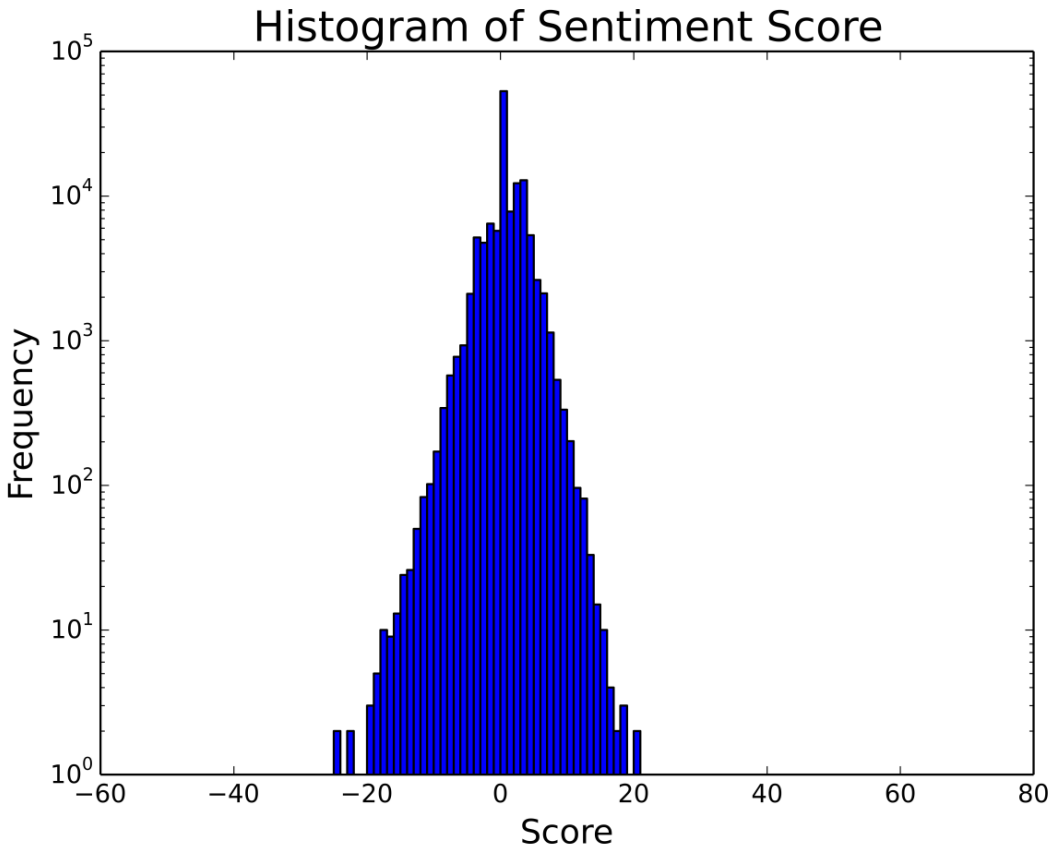

Figure 3 Sentiment Score of Twitter Text Distribution

那么我们接下来，便很自然地想到这样的情绪分数在不同的登陆途径上的分布是怎样的呢？情绪分数和登陆途径之间是否有着某种模式呢？

从图中（Figure 4）我们可以发现，用户量相对较多的 Twitter for iphone 和 Twitter for Android 的情感得分均值分别为 0.3128（标准误为 0.0106）和 0.2945（标准误为 0.0187），都没有达到平均水平 0.3944。而来自 Unfollowers.me 的用户反而得分为负值 -0.5861（标准误为 0.0271）。这个或许存在一定的原因亟待细究。得分最高的是来自 Google 的 Twitter 用户，得分为 1.275（标准误为 0.1727）。Google 用户在传播积极情绪上有着不错的效果。另外，来自于 Instagram 用户的得分也都比较高。

4 总结

在这个开放项目中，我在 Twitter 上采集了 1 个小时的用户发布或转发活动之后，进行了简单的情感刻画。这种情感刻画仅仅针对词汇的消极/积极性质，并不探究语义分析。我们看到了来自不同 Twitter 登陆渠道的用户，情感分布不同。但是，总体而言，采集数据的那段时间中，Twitter 上所反映的其用户的情感是略微偏向与正向的、积极的。另外，Twitter 用户中，使用 iPhone 登陆的用户占据了相当大的比例。

这次的情感分析，我只采用了每条 Twitter 自身的信息。更一般地，我们还可以将用户的信息包括进来，寻找诸如用户的情感状态与用户的其他资料（如性别、个性化主页、关注/被关注

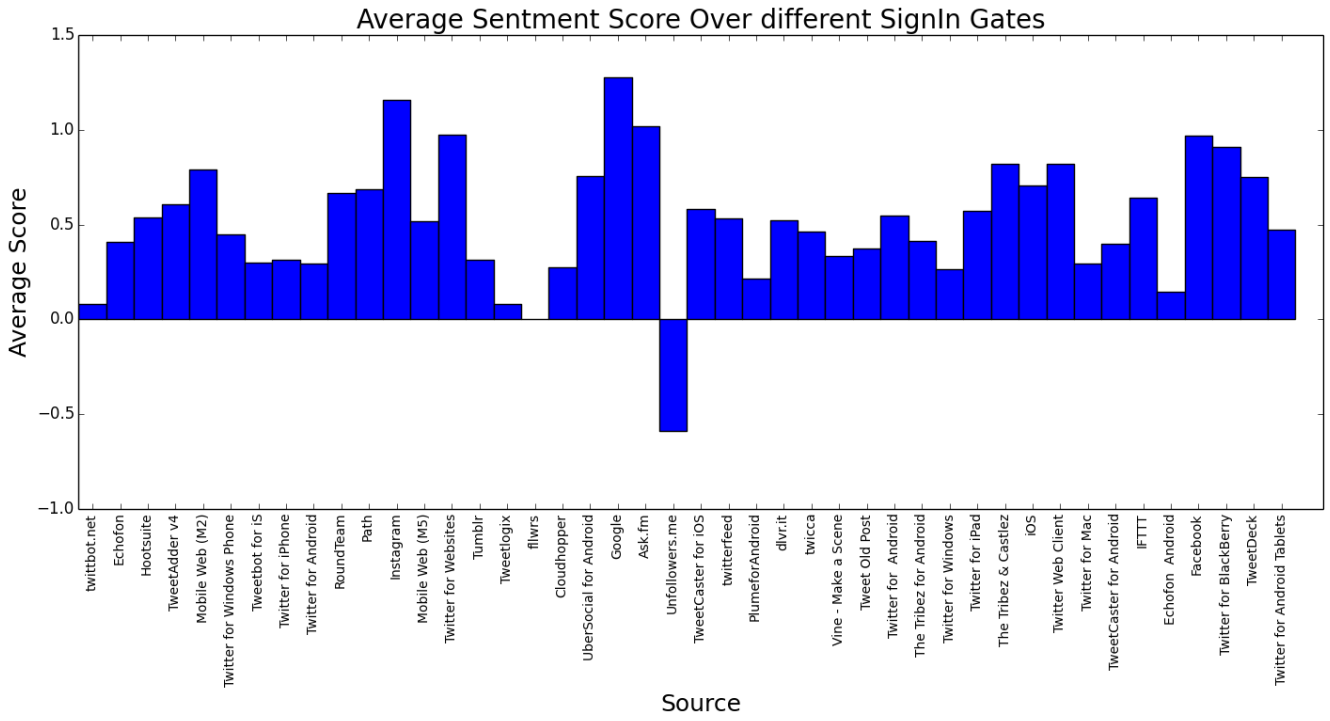

Figure 4 Average Sentiment Score Over different Sign in Source

情况、地理信息等等）之间的联系。我们可以从宏观层面来观察舆情走向，也可以从微观角度来侦测个人的情感波动。

References

[1] 若. 离, “Twitter 每月 2.15 亿活跃用户 面临增长问题,” 04 10 2013. [联机]. Available: http://mobile.163.com/13/1004/08/9AB33B220011671M.html.

[2] 聪. 吉, "新浪微博核心数据:月活跃用户数 1.2 亿," 15 03 2014. [Online]. Available: http://it.sohu.com/20140315/n396647661.shtml.

[3] 钟, 瑛; 刘, 利芳;, "微博传播的舆论影响力," 21 02 2013. [Online]. Available: http://media.people.com.cn/n/2013/0221/c40628-20557147.html.

[4] Ampofo, Lawrence; Collister, Simon; O'Loughlin, Ben; Chadwick, Andrew;, "Text Mining and Social Media: When Quantitative Meets Qualitative, and Software Meets Humans," in *SAGE Publications Ltd*, 2014.

[5] Deitrick, William; Hu, Wei;, "Mutually Enhancing Community Detection and Sentiment," *Journal of Data Analysis and Information Processing,* pp. 19-29, 2013.

[6] 蒋, 盛益; 麦, 智凯; 庞, 观松; 吴, 美玲; 王, 连喜;, "微博信息挖掘技术研究综述," *图书情报工作,* pp. 136-142, 9 2012.

[7] Wang, Hao; Can, Dogan; Kazemzadeh, Abe; Bar, Francois; Narayanan, Shrikanth;, "A System for Real-time Twitter Sentiment Analysis of," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, Jeju, Republic of Korea, 2012.

[8] Bollen, Johan; Pepe, Alberto; Mao, Huina;, "Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena," in *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, Barcelona, Spain, 2011.

[9] F. A. Nielsen, AFINN, Informatics and Mathematical Modelling, Technical University of Denmark, 2011.

Appendix:

```
input_file = "output.txt"
#output_file = "simplified_output.txt"

#!/usr/bin/python2.7
# -*- coding: utf-8 -*-

import os
import re
import json, inspect
#import pylab, pandas
#from nltk import *

os.getcwd()
os.chdir("C:\Users\Administrator\Desktop\WorkShop")

# create score_dictionary;
afinn_dic = {}
afinn = open("AFINN-111.txt","r")
afinn_list = afinn.readlines()
afinn.close()
for term_i in range(len(afinn_list)):
        afinn_tmp = afinn_list[term_i].strip().split('\t')
        afinn_dic[afinn_tmp[0]] = int(afinn_tmp[1])
```

```
import numpy;
#from pylab import *;
#from matplotlib import pyplot
i = 0; j = 0;
inputfile = open(input_file,"r")
data = [];user_id=[];
features_twitter = [u'id_str',u'text',u'source',u'coordinates',u'retweet_count',u'favorite_count',u'lang'];
features_user =
[u'id_str',u'location',u'time_zone',u'geo_enabled',u'followers_count',u'friends_count',u'listed_count',u'c
reated_at',u'favourites_count',u'timezone',u'statuses_count',u'lang',u'profile_backgroud_color',u"profil
e_sidebar_border_color",u'profile_sidebar_fill_color',u'profile_text_color',u'profile_use_background_i
mage']
twitters = []
twitters_data = {}
twitter_number = 0;feature_number=0;
for eachf in features_twitter:
    feature_number = feature_number+1;
    twitters_data[str(eachf)] = [];
for eachline in inputfile:
    i = i+1;
    if numpy.mod(i,100000) == 0:
        print i;
    eachline.encode('ascii','ignore')
    twitter_number = twitter_number+1;
    twitter_content = [];
    dic = json.loads(eachline);
    data.append(dic);
    for feature in features_twitter:
        try:
```

```
            twitter_content.append(dic[feature])
            twitters_data[feature].append(dic[feature])
        except KeyError:
            twitter_content.append("Emptyyyyy")
            twitters_data[feature].append("Emptyyyyy")
    twitters.append(twitter_content)
#twitters_array = numpy.array(twitters)
#twitters_data_array = numpy.array(twitters_data)
inputfile.close()



print "Read Finished!";
print "Start Write!"

feature_noempty = {}
feature_noempty['score'] = []
hh = 0
for eachFeature in features_twitter:
    hh = hh+1
    print hh;
    #feature_noempty[eachFeature] = [x for x in twitters_data[eachFeature] if x !="Emptyyyyy"]
    feature_noempty[eachFeature] = twitters_data[eachFeature]
    for i in range(len(feature_noempty[eachFeature])):
        try:
            if eachFeature == "source":
                tmp = feature_noempty[eachFeature][i]
                try:
                    feature_noempty[eachFeature][i] = tmp[tmp.index('>')+1:tmp.index('<',tmp.index('>'))]
                except:
```

```
            feature_noempty[eachFeature][i] = tmp[tmp.index('<')+1:tmp.index('>',tmp.index('<'))]
        tmp2 = feature_noempty[eachFeature][i].encode('ascii','ignore')
        feature_noempty[eachFeature][i] = tmp2.strip()
        #print eachFeature
        if eachFeature == 'text':
          #print "fuck"
          score = 0;
          txt = re.sub('[^A-Za-z0-9]+',' ',feature_noempty[eachFeature][i]).lower().strip().split(' ');
          for eachWord in txt:
            try:
              score = score + afinn_dic[eachWord];
            except:
              pass
          feature_noempty["score"].append(score)
      except:
        pass

'''dump the data dic into an extra file'''
json.dump(feature_noempty, open("feature_empty.txt",'w'));



import os
import re
import json, inspect, pandas, numpy
#from nltk import *
#import numpy;
from pylab import *;
from matplotlib import pyplot
from scipy import stats
```

```
import matplotlib



os.getcwd()
os.chdir("C:\Users\Administrator\Desktop\WorkShop")

datause = json.loads(open('feature_empty.txt','r').read())
dataframe_twitter = pandas.DataFrame(datause)



def FreqTable(feature):
    key = []
    value = []
    for iterm in list(set(feature)):
        iter_var = 0;
        key.append(str(iterm))
        for iterm_m in feature:
            if iterm_m == iterm:
                iter_var = iter_var+1;
        value.append(iter_var)
    return value,key



def BarChart(feature,name="",bottom_height=0.05):
    #frame = inspect.currentframe()
    #args, varargs, keywords, values = inspect.getargvalues(frame)
    #print args, varargs, keywords
    value,key = FreqTable(feature)
    #return (value,key)
```

```
    fig = pyplot.figure()
    pyplot.title(name+" Distribution", fontsize=20)
    pyplot.ylabel("Frequency",fontsize=16)
    pyplot.bar(numpy.arange(len(key)),value,width=1.0)
    pyplot.xticks(numpy.arange(len(value))+0.4,key,rotation=90, size='medium')
    pylab.subplots_adjust(bottom = 0.01)
    pyplot.savefig(name+".png", dpi=1000)
    pyplot.show()



'''
Language Distribution
'''
data_lang = [x for x in datause['lang'] if x !='Emptyyyyy']
#BarChart(data_lang,'Language')



'''
SignIn Source Distribution
'''
data_source = dataframe_twitter[dataframe_twitter['lang']=='en']['source']

'''
['Echofon', 'Hootsuite', 'TweetAdder v4', 'Mobile Web (M2)', 'Twitter for Windows Phone',
'Tweetbot for iS', 'Twitter for iPhone', 'Twitter for Android', 'RoundTeam', 'Instagram',
'Mobile Web (M5)', 'Twitter for Websites', 'Tweetlogix', 'Cloudhopper', 'Unfollowers.me',
'TweetCaster for iOS', 'twitterfeed', 'PlumeforAndroid', 'dlvr.it', 'Twitter for  Android',
'Twitter for iPad', 'iOS', 'Twitter Web Client', 'Twitter for Mac',
```

```
'TweetCaster for Android', 'IFTTT', 'Facebook', 'Twitter for BlackBerry', 'TweetDeck',
'Twitter for Android Tablets']
'''
def BarChart(feature,name="",bottom_height=0.3):
    #frame = inspect.currentframe()
    #args, varargs, keywords, values = inspect.getargvalues(frame)
    #print args, varargs, keywords
    value,key = FreqTable(feature)
    key_new = [key[x] for x in range(len(key)) if value[x] >= 50 ]
    value_new = [x for x in value if x >= 50]
    fig = pyplot.figure()
    pyplot.title(name+" Distribution",fontsize=20)
    pyplot.ylabel("Frequency",fontsize=18)
    pyplot.bar(numpy.arange(len(key_new)),value_new,width=0.8)
    pyplot.xticks(numpy.arange(len(value_new))+0.4,key_new,rotation=90, size='small')
    subplots_adjust(bottom = bottom_height)
    pyplot.savefig(name+".png", dpi=1000)
    pyplot.show()

#[elem for elem in li if li.count(elem) == 1]

#BarChart(data_source,'SignIn Gate',0.34)

'''
histogram of Sentiment score
'''
#fig = pyplot.figure()
#x = list(dataframe_twitter[dataframe_twitter['lang'] == 'en']['score']);
#pyplot.hist(x,bins=107,log=True)
```

```
#pyplot.ylabel('Frequency',fontsize=16);pyplot.xlabel('Score',fontsize=16);

#pyplot.title('Histogram of Sentiment Score',fontsize=20)

#pyplot.savefig("sentiment_score_dist.png", dpi=1000)

#show()

#print 't-statistic = %6.3f pvalue = %6.4f' %  stats.ttest_1samp(x, 0)

'''t-statistic = 49.995 pvalue = 0.0000'''


'''

score exploration by valid source

'''

#value,key = FreqTable(data_source)

#key_new = [key[x] for x in range(len(key)) if value[x] >= 50 ]

#new_frame = numpy.array(dataframe_twitter)

#source_score = [mean(new_frame[(new_frame[:,6] == ss) & (new_frame[:,3] == 'en'),5]) for ss in
key_new]

#

def BarChart_source_score(name="Avg Score by Source",bottom_height=0.3):

    fig = pyplot.figure()

    pyplot.title("Average Sentment Score Over different SignIn Gates",fontsize=20)

    pyplot.ylabel("Average Score",fontsize=18)

    pyplot.xlabel('Source',fontsize=18)

    pyplot.bar(numpy.arange(len(key_new)),source_score,width=1)

    pyplot.xticks(numpy.arange(len(source_score))+0.5,key_new,rotation=90, size='small')

    subplots_adjust(bottom = bottom_height)

    pyplot.savefig(name+".png", dpi=1000)

    pyplot.show()


#source_score_ttest = [stats.ttest_1samp(new_frame[(new_frame[:,6] == ss) & (new_frame[:,3] ==
'en'),5],0) for ss in key_new]
```

```
#for each in source_score_ttest:
#           print "t=%6.3f p=%6.4f" % each;
'''
standard_error
'''
#std = [std(new_frame[(new_frame[:,6] == ss) & (new_frame[:,3] == 'en'),5]) for ss in key_new]
#df = [len(new_frame[(new_frame[:,6] == ss) & (new_frame[:,3] == 'en'),5])-1 for ss in key_new]
#den = sqrt(df)
#se = std/den
'''
twitter_favoriated
'''
#fav = dataframe_twitter[dataframe_twitter['lang']=='en']['favorite_count']


'''
coordinates
'''
#coord = dataframe_twitter[dataframe_twitter['lang']=='en']['coordinates']
'''
i = 0;
coord_new = [];
for each in coord:
   if each != None:
      coord_new.append(tuple(each['coordinates']));
   i = i+1;
mc = matrix(coord_new)
mc = numpy.array(coord_new)
'''
'''
```

```
dic_new = {}
x = list(dataframe_twitter[dataframe_twitter['lang'] == 'en']['score'])
i = 0;j=0;
for each_i in range(len(coord)):
    if coord[each_i] != None:
        try:
                        dic_new[tuple(coord[each_i]['coordinates'])] = x[each_i];
            i = i +1;
    except KeyError:
            print j;
    j = j+1;

'''
```