现代程序设计第三次作业

实现过程

导包和加载入情绪词

import matplotlib.pyplot as plt  
import jieba  
import numpy as np  
jieba.load\_userdict(r'Anger makes fake news viral online-data&code/data/emotion\_lexicon/anger.txt')  
jieba.load\_userdict(r'Anger makes fake news viral online-data&code/data/emotion\_lexicon/disgust.txt')  
jieba.load\_userdict(r'Anger makes fake news viral online-data&code/data/emotion\_lexicon/fear.txt')  
jieba.load\_userdict(r'Anger makes fake news viral online-data&code/data/emotion\_lexicon/joy.txt')  
jieba.load\_userdict(r'Anger makes fake news viral online-data&code/data/emotion\_lexicon/sadness.txt')  
import jieba.analyse  
import re

利用闭包读取情绪词

def get\_path(file\_path):  
 file\_path = r'Anger makes fake news viral online-data&code/data/emotion\_lexicon/'  
 def read\_exact(emotion):  
 nonlocal file\_path  
 file\_path += emotion  
 return file\_path  
 return read\_exact

读取文件

def read\_emotion(file):  
 with open(file,'r',encoding='utf-8') as f:  
 words = f.read()  
 words\_list = words.split('\n')  
 return words\_list  
  
  
def read\_txt(file):  
 with open(file,'r',encoding='utf-8') as f:  
 sentences = f.read()  
 text\_list = sentences.split('\n')  
 return text\_list  
  
def read\_stopwords(file):  
 with open(file,'r',encoding='utf-8') as f:  
 stopwords = f.read()  
 stopwords\_list = stopwords.split('\n')  
 return stopwords\_list

数据清洗

def clean(text\_list):  
 for i in range(len(text\_list)):  
 text\_list[i] = re.sub(r"(回复)?(//)?\s\*@\S\*?\s\*(:| |$)", " ", text\_list[i]) # 去除正文中的@和回复/转发中的用户名  
 text\_list[i] = re.sub(r"\[\S+\]", "", text\_list[i]) # 去除表情符号  
 text\_list[i] = re.sub(r"#\S+#", "", text\_list[i]) # 保留话题内容  
 URL\_REGEX = re.compile(  
 r'(?i)\b((?:https?://|www\d{0,3}[.]|[a-z0-9.\-]+[.][a-z]{2,4}/)(?:[^\s()<>]+|\(([^\s()<>]+|(\([^\s()<>]+\)))\*\))+(?:\(([^\s()<>]+|(\([^\s()<>]+\)))\*\)|[^\s`!()\[\]{};:\'".,<>?«»“”‘’]))',  
 re.IGNORECASE)  
 text\_list[i] = re.sub(URL\_REGEX, "", text\_list[i]) # 去除网址  
 text\_list[i] = text\_list[i].replace("转发微博", "") # 去除无意义的词语  
 text\_list[i] = re.sub(r"\s+", " ", text\_list[i]) # 合并正文中过多的空格  
 return text\_list

分词处理

def word\_cut(text\_list):  
 text\_cut\_list = []  
 for i in range(len(text\_list)):  
 text\_cut = jieba.lcut(text\_list[i])  
 text\_cut\_list.append(text\_cut)  
 return text\_cut\_list  
  
def stop\_words(text\_cut\_list,stopwords\_list):  
 text\_stop\_list = []  
 for i in range(len(text\_cut\_list)):  
 text\_stop = []  
 for j in range(len(text\_cut\_list[i])):  
 if (text\_cut\_list[i][j] in stopwords\_list) or text\_cut\_list[i][j] == ' ':  
 pass  
 else:  
 text\_stop.append(text\_cut\_list[i][j])  
 text\_stop\_list.append(text\_stop)  
 return text\_stop\_list

获取文本中的时间位置数据

def get\_time(text\_list):  
 time\_list = []  
 location\_list = []  
 for i in range(len(text\_list)):  
 #for i in range(6):  
 #length = len(text\_list[i])  
 mark = text\_list[i].rfind('[')  
 location\_list.append(text\_list[i][mark:])  
 time\_list.append(text\_list[i][mark - 31:mark])  
 text\_list[i] = text\_list[i][0:mark - 31]  
 #print(location\_list)  
 #print(time\_list)  
 #print(text\_list[0])  
  
 return text\_list,time\_list, location\_list

分析每条文本的情绪构成

def emotion\_analsis(text\_stop\_list,emotion\_list):  
 emotion\_vector\_list = []  
 emotion\_value\_list = []  
 for i in range(len(text\_stop\_list)):  
 emotion\_vector = {'anger': 0, 'disgust': 0, 'fear': 0, 'joy': 0, 'sadness': 0}  
 for word in text\_stop\_list[i]:  
 if word in emotion\_list[0]:  
 emotion\_vector['anger'] += 1  
 elif word in emotion\_list[1]:  
 emotion\_vector['disgust'] += 1  
 elif word in emotion\_list[2]:  
 emotion\_vector['fear'] += 1  
 elif word in emotion\_list[3]:  
 emotion\_vector['joy'] += 1  
 elif word in emotion\_list[4]:  
 emotion\_vector['sadness'] += 1  
 else:  
 pass  
 max = 0  
 emotion\_value = 'none'  
 for key in emotion\_vector:  
 if emotion\_vector[key] > max:  
 max = emotion\_vector[key]  
 emotion\_value = key  
 emotion\_vector\_list.append(emotion\_vector)  
 emotion\_value\_list.append(emotion\_value)  
 return emotion\_vector\_list,emotion\_value\_list

情绪的小时模式

def emotion\_hourmode(time\_list,emotion\_value\_list):  
 emotion\_hour\_list = [{'anger': 0, 'disgust': 0, 'fear': 0, 'joy': 0, 'sadness': 0} for i in range(24)]  
 #print(emotion\_hour\_list)  
 emotion\_value\_list.pop()  
 for i in range(len(emotion\_value\_list)):  
 hour = int(time\_list[i][11:13])  
 #print(emotion\_value\_list[hour])  
 if emotion\_value\_list[i] in emotion\_hour\_list[hour]:  
 emotion\_hour\_list[hour][emotion\_value\_list[i]]+=1  
 return emotion\_hour\_list

情绪的空间模式

def emotion\_locationmode(location\_list, emotion\_value\_list):  
 location\_x\_list = []  
 location\_y\_list = []  
 location\_emotion = []  
 color\_list = ['coral','khaki','aquamarine','deepskyblue','violet']  
 emotion\_list = ['anger','disgust','fear','joy','sadness']  
 #print(location\_list)  
 for i in range(len(emotion\_value\_list)-1):  
 location\_x\_list.append(eval(location\_list[i])[0])  
 location\_y\_list.append(eval(location\_list[i])[1])  
 location\_emotion.append(emotion\_value\_list[i])  
 for i in range(5):  
 x = []  
 y = []  
 for j in range(len(emotion\_value\_list)-1):  
 if emotion\_list[i] == location\_emotion[j]:  
 x.append(location\_x\_list[j])  
 y.append(location\_y\_list[j])  
 plt.scatter(x, y, s=None, c=color\_list[i], marker=None, cmap=None, norm=None, vmin=None, vmax=None, alpha=None, linewidths=None, edgecolors=None, plotnonfinite=False, data=None)  
 plt.title('the location mode of '+emotion\_list[i])  
 plt.show()

主函数

def main():  
 text\_list = read\_txt('weibo.txt')  
 stopwords\_list = read\_stopwords('stopwords\_list.txt')  
 #print(stopwords\_list)  
 text\_list = clean(text\_list)  
 text\_list, time\_list ,location\_list = get\_time(text\_list)  
 text\_cut\_list = word\_cut(text\_list)  
 text\_stop\_list = stop\_words(text\_cut\_list,stopwords\_list)  
  
  
 emotion\_list = []  
 emotion\_list.append(read\_emotion(get\_path('')('anger.txt')))  
 emotion\_list.append(read\_emotion(get\_path('')('disgust.txt')))  
 emotion\_list.append(read\_emotion(get\_path('')('fear.txt')))  
 emotion\_list.append(read\_emotion(get\_path('')('joy.txt')))  
 emotion\_list.append(read\_emotion(get\_path('')('sadness.txt')))  
 emotion\_vector\_list,emotion\_value\_list = emotion\_analsis(text\_stop\_list,emotion\_list)  
  
 ans = count(emotion\_value\_list)  
 print(ans)  
  
 #for i in range(len(text\_list)):  
 for i in range(1):  
 print('-----------------------------------------------------------------------------------------------')  
 print('num:',i)  
 print('微博原文：',text\_list[i])  
 print('提取分词：',text\_stop\_list[i])  
 print('情绪词向量：',emotion\_vector\_list[i])  
 print('主要情绪类型：',emotion\_value\_list[i])  
 print('微博发送时间：',time\_list[i])  
 print('微博发送地点',location\_list[i])  
 print('-----------------------------------------------------------------------------------------------')  
  
 emotion\_hour\_list = emotion\_hourmode(time\_list,emotion\_value\_list)  
 print(emotion\_hour\_list)  
  
 x = np.arange(24)  
 type\_list = ['anger','disgust','fear','joy','sadness']  
 for i in range(5):  
 hour\_draw\_list = [0 for i in range(24)]  
 for j in range(24):  
 hour\_draw\_list[j] = emotion\_hour\_list[j][type\_list[i]]  
 plt.figure(figsize=(10, 5))  
 plt.plot(x,hour\_draw\_list,'black',label=type\_list[i])  
 plt.title(f'the time about {type\_list[i]}')  
 plt.grid()  
 plt.show()  
  
 emotion\_locationmode(location\_list, emotion\_value\_list)  
  
main()

部分结果展示

情绪总数统计

{'none': 5490, 'disgust': 377, 'joy': 3272, 'fear': 151, 'anger': 540, 'sadness': 771}

每条情绪分析结果

num: 0

微博原文： 分享图片 我在这里:

提取分词： ['分享', '图片']

情绪词向量： {'anger': 0, 'disgust': 0, 'fear': 0, 'joy': 0, 'sadness': 0}

主要情绪类型： none

微博发送时间： Fri Oct 11 21:25:07 +0800 2013

微博发送地点 [39.88293, 116.37024]

时间模式统计

[{'anger': 33, 'disgust': 26, 'fear': 4, 'joy': 185, 'sadness': 32}, {'anger': 9, 'disgust': 3, 'fear': 3, 'joy': 43, 'sadness': 24}, {'anger': 1, 'disgust': 6, 'fear': 4, 'joy': 19, 'sadness': 9}, {'anger': 4, 'disgust': 0, 'fear': 0, 'joy': 11, 'sadness': 4}, {'anger': 6, 'disgust': 0, 'fear': 3, 'joy': 20, 'sadness': 1}, {'anger': 4, 'disgust': 0, 'fear': 0, 'joy': 11, 'sadness': 8}, {'anger': 6, 'disgust': 4, 'fear': 0, 'joy': 31, 'sadness': 12}, {'anger': 4, 'disgust': 15, 'fear': 14, 'joy': 70, 'sadness': 29}, {'anger': 32, 'disgust': 24, 'fear': 9, 'joy': 109, 'sadness': 40}, {'anger': 26, 'disgust': 11, 'fear': 0, 'joy': 170, 'sadness': 34}, {'anger': 24, 'disgust': 17, 'fear': 2, 'joy': 148, 'sadness': 29}, {'anger': 28, 'disgust': 13, 'fear': 6, 'joy': 101, 'sadness': 19}, {'anger': 21, 'disgust': 15, 'fear': 4, 'joy': 165, 'sadness': 53}, {'anger': 19, 'disgust': 21, 'fear': 5, 'joy': 158, 'sadness': 35}, {'anger': 19, 'disgust': 3, 'fear': 1, 'joy': 185, 'sadness': 38}, {'anger': 16, 'disgust': 21, 'fear': 11, 'joy': 135, 'sadness': 32}, {'anger': 38, 'disgust': 21, 'fear': 8, 'joy': 145, 'sadness': 28}, {'anger': 27, 'disgust': 17, 'fear': 7, 'joy': 120, 'sadness': 46}, {'anger': 24, 'disgust': 18, 'fear': 6, 'joy': 150, 'sadness': 57}, {'anger': 33, 'disgust': 26, 'fear': 8, 'joy': 182, 'sadness': 43}, {'anger': 55, 'disgust': 19, 'fear': 13, 'joy': 243, 'sadness': 28}, {'anger': 31, 'disgust': 17, 'fear': 1, 'joy': 201, 'sadness': 66}, {'anger': 24, 'disgust': 17, 'fear': 14, 'joy': 203, 'sadness': 32}, {'anger': 56, 'disgust': 63, 'fear': 28, 'joy': 467, 'sadness': 72}]

绘图









