

现代程序设计第3周作业

谢奕飞 20377077

数据读取函数ReadData()

```
1  def ReadData(file_path):
2      '''
3      数据读取函数
4      传入txt文件绝对路径
5      返回列表(列表每个元素为time、location、text、emotion构成的字典)
6      '''
7      f=open(file_path,'r',encoding='utf8')
8      data=[]
9      f.readline()#去表头
10     for line in f.readlines():
11         text=''
12         time=''
13         location=[]
14         #处理text
15         message=line.split()
16         for each in message[2:-7]:
17             text+=' '+each
18         #处理location
19         location.append(float(message[0][1:-2]))
20         location.append(float(message[1][1:-2]))
21         #处理time
22         for each in message[-6:]:
23             time+=' '+each
24         t=dt.datetime.strptime(time,' %a %b %d %H:%M:%S %z %Y')
25         data.append({'time':t,'location':location,'text':text,'emotion':'\n
one'})
26     f.close()
27     return data
28
29
30 def main():
31     '''
32     测试函数
33     '''
34     file_path='D:\Project\Python\week3weibo\weibo1.txt'
35     data=ReadData(file_path)
36     print(data)
```

```

d:\Project\Python - VS Code 控制台
[{'time': datetime.datetime(2013, 10, 11, 22, 8, 28, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.6784, 116.10992], 'text': '也许生活就像一本掉了页的黄历,没有人会把它粘好并一页一页的回忆。我在:http://t.cn/zRGaM7d', 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 23, 31, 7, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.67847, 116.15264], 'text': '今夜到明天白天有点想你,预计下午将持续想你,受延长低情绪影响,傍晚将转为大到暴想,心情由此将降低五度,预计此类天气将持续到见到你为止。@Sandy-英俊哥 米修 搜麻奇~~~[亲亲][亲亲][亲亲] 幸好还有酸龟龟陪我[悲伤] 我在:http://t.cn/zRG9LQA', 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 23, 54, 47, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.67692, 116.2939], 'text': '【抓狂】像跟个定时炸弹聊天[衰][衰][衰]我在这里:http://t.cn/z8A4AnU', 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 23, 24, 48, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.68807, 116.3115], 'text': '【垂直电商低成本运营的三路九招】中国垂直电商逐步开始从规模导向转为利润导向。电商都在节衣缩食,首当其冲就是对营销费用也精打细算。第一路:流量导入第二路:用户互动第三路:资源整合...下图为您呈现三路九招式低成本运行。我在:http://t.cn/zRCKdx5', 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 16, 46, 37, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.67287, 116.34996], 'text': '我用#智行火车票#,成功秒杀10月08日北京西-广州东的硬座。http://t.cn/zHeqZgx 我在:http://t.cn/zRGbQHP', 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 10, 24, 8, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.68097, 116.37997], 'text': '我们都有看不开的时候,总有冷落自己的举动,但是我我一定会提醒自己,如果还有明天,我们都有伤心的时候,总不在乎这种感受,但是我会提醒自己珍惜每次感动..... 如果还有明天..... 如果没有明天..... 《如果还有明天》 我在:http://t.cn/zRZ3r5T', 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 23, 2, 33, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.67982, 116.41157], 'text': '翻了翻以前的照片,想了想以前的事儿。笑过哭过,吵过闹过,爱过恨过。其实,挺美好的,曾经,有.....。就这样,让它美好下去吧。 我在:http://t.cn/zRGoeNa', 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 23, 25, 37, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.69015, 116.4346], 'text': '【#易迅免费送iphone5c#】免费送iphone5c来啦,易迅无线惊喜无限,iphone5c免费送,红米购买权免费抽,总有一款适合你!小伙伴们速度来试下运气!今天,你的人品爆发了没! http://t.cn/z8sfxIk 我在:http://t.cn/zRGzLAY', 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 23, 38, 48, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.67475, 116.52044], 'text': '[爱你]王诗龄和kimi。。。有没有口水流干了死翘的啊[可怜]我快了 我在这里:http://t.cn/z8A4ylE', 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 22, 30, 16, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.67639, 116.52782], 'text': '不行了<爸爸去哪儿>这节目看不下去了。。。超想生孩子。我先在沙发里假装睡着了听一耳朵吧[抓狂]好喜欢王诗龄和kimi 我在:http://t.cn/zRGS6Vx', 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 19, 3, 36, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.6846, 116.5931], 'text': '十天培训结束第一天进组上班感觉好压抑,以后下班早不了,感觉干不长呀 我在这里:http://t.cn/z8AxVd7'}]

```

function: 传入文件绝对路径, 将每行数据提取为包含time、location、emotion、text的键的字典(emotion初始化为'none'), 返回以该字典为元素的列表

数据清洗函数CleanData()

```

1 def CleanData(data):
2     '''
3     清洗数据, 去除重复值和噪声(url
4     传入列表data(列表每个元素为time、location、text、emotion构成的字典)
5     返回清洗后的列表(text的值为词构成的列表)
6     '''
7     res=[]
8     url=re.compile(r'[http|https]*://[a-zA-Z0-9.?/&=:]*')#取url
9     for each in data:
10         each['text']=re.sub(url, '', each['text'])#去url
11         if each not in res:#去重
12             res.append(each)
13     return res
14
15 def main():
16     '''
17     测试函数
18     '''
19     file_path='D:\Project\Python\week3weibo\weibo1.txt'
20     data=ReadData(file_path)
21     data=CleanData(data)
22     print(data)

```

```

d:\Project\Python - VS Code 控制台
[{'time': datetime.datetime(2013, 10, 11, 22, 8, 28, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.6784, 16.10992], 'text': '也许生活就像一本掉了页的黄历,没有人会把它粘好并一页一页的回忆。我在:', 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 23, 31, 7, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.67847, 16.15264], 'text': '今夜到明天白天有点想你,预计下午将持续想你,受延长低情绪影响,傍晚将转为大到暴想,心情由此将降低五度,预计此类天气将持续到见到你为止。@Sandy-英俊哥 米修 搜麻奇~~[亲亲][亲亲][亲亲] 幸好还有酸龟龟陪我[悲伤] 我在:', 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 23, 54, 47, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.67692, 16.2939], 'text': '[抓狂][抓狂]像跟个定时炸弹聊天[衰][衰][衰] 我在这里:', 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 23, 24, 48, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.68807, 16.3115], 'text': '【垂直电商低成本运营的三路九招】中国垂直电商逐步开始从规模导向转为利润导向。电商都在节衣缩食,首当其冲就是对营销费用也精打细算。第一路:流量导入第二路:用户互动第三路:资源整合...下图为您呈现三路九招式低成本运行。我在:', 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 16, 46, 37, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.67287, 16.34996], 'text': '我用#智行火车票#,成功秒杀10月08日北京西-广州东的硬座。我在:', 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 10, 24, 8, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.68097, 16.37997], 'text': '我们都有看不开的时候,总有冷落自己的举动,但是我我一定会提醒自己,如果还有明天,我们都有伤心的时候,总不在乎这种感受,但是我会提醒自己珍惜每次感动..... 如果还有明天..... 如果没有明天..... 《如果还有明天》 我在:', 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 23, 2, 33, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.67982, 16.41157], 'text': '翻了翻以前的照片,想了想以前的事儿。笑过哭过,吵过闹过,爱过恨过。其实,挺美好的。曾经,有.....。就这样,让它美好下去吧。我在:', 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 13, 25, 37, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.69015, 16.4346], 'text': '【#易迅免费送iphone5c#】免费iphone5c来啦,易迅无线惊喜无限,iphone5c免费送,红米购买权免费抽,总有一款适合你!小伙伴们速度来试下运气!今天,你的人品爆发了没! 我在:', 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 23, 38, 48, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.67475, 16.52044], 'text': '[爱你]王诗龄和kimi。。有没有口水流干了死翘翘的啊[可怜]我快了 我在这里:', 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 22, 30, 16, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.67639, 16.52782], 'text': '不行了<爸爸去哪儿>这节目看不下去了。。。超想生孩子。我先歪沙发里假装睡着了听一耳朵吧[抓狂]好喜欢王诗龄和kimi 我在:', 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 19, 3, 36, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.6846, 16.5931], 'text': '十天培训结束第一天进组上班感觉好压抑,以后下班早不了,感觉干不长呀 我在这里:', 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 20, 49, 8, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.68124, 16.69732], 'text': '我正在搜狗手机输入法的“超级键手”比赛中抽奖,大奖拿到手软,快来试试手气!'}]
```

function: 将data的每个元素（字典）的'text'值中的url替换为空，对比data每个元素，剔除重复项

文本分割函数CutText()

```

1 def CutText(data,dict_path):
2     '''
3     将data中的text分词
4     传入列表data(列表每个元素为time、location、text、emotion构成的字典)和词典地址
5     返回text分词后的列表
6     '''
7     f=open(dict_path,'r',encoding='utf8')
8     stop_words = [line.strip() for line in f.readlines()]
9     jieba.load_userdict(dict_path)#载入词典
10    for each in data:
11        words=[]
12        for word in list(jieba.cut(each['text'])):
13            if word not in stop_words:#过滤停用词
14                words.append(word)
15        each['text']=words
16    f.close()
17    return data
18
19 def main():
20     '''
21     测试函数
22     '''
23     file_path='D:\Project\Python\week3weibo\weibo1.txt'
24     dict_path='D:\Project\Python\week3weibo\stopwords_list.txt'
25     data=ReadData(file_path)
26     clean_data=CleanData(data)
27     seg_data=CutText(clean_data,dict_path)
28     print(seg_data)
```

```
d:\Project\Python - VS Code 控制台
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\NO_THA 1\AppData\Local\Temp\jieba.cache
Loading model cost 0.566 seconds.
Prefix dict has been built successfully.
[{'time': datetime.datetime(2013, 10, 11, 22, 8, 28, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.6784, 16.10992], 'text': [' ', '也许', '生活', '一本', '掉', '页', '黄页', '人会', '粘', '一页', '一页', '回忆', ''], 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 23, 31, 7, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.67847, 16.15264], 'text': [' ', '今夜', '明天', '白天', '想', '预计', '持续', '想', '想', '受', '延长', '低', '情绪', '影响', '傍晚', '转为', '大到', '暴想', '心情', '降低', '五度', '预计', '此类', '天气', '持续', '见到', 's', 'y', '英俊', '哥', '米', '修', '搜麻奇', '一', '一', '亲亲', '亲亲', '亲亲', '幸好', '酸角', '角', '陪', '悲伤', ''], 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 23, 54, 47, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.67692, 16.2939], 'text': [' ', '抓狂', '抓狂', '定时炸弹', '聊天', '聊天', '衰', '衰', '衰', '衰', ''], 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 23, 24, 48, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.68807, 16.3115], 'text': [' ', '垂直', '电商', '低成本', '运营', '三路', '九招', '中国', '垂直', '电商', '规模', '导向', '转为', '利润', '导向', '电商', '节衣缩食', '首当其冲', '营销', '费用', '精打细算', '第一路', '流量', '导入', '第二路', '用户', '互动', '第三路', '资源整合', '下图', '呈现', '三路', '招式', '低成本', '运行', ''], 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 16, 46, 37, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.67287, 16.34996], 'text': [' ', '我用', '智行', '火车票', '成功', '秒杀', '北京西', '广州', '东', '硬座', ''], 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 10, 24, 8, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.68097, 16.37997], 'text': [' ', '不开', '总有', '冷落', '举动', '提醒', '明天', '伤心', '总', '不在乎', '感受', '我会', '提醒', '珍惜', '每次', '感动', '明天', '明天', ''], 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 23, 2, 3, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.67982, 16.41157], 'text': [' ', '翻', '翻', '照片', '想', '想', '事儿', '笑', '哭', '吵过', '闹过', '爱过', '恨', '挺', '美好', '美好', ''], 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 13, 25, 37, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.69015, 16.4346], 'text': [' ', '易迅', '免费送', 'iph', 'c', '免费', 'iph', '5c', '易迅', '无线', '惊喜', '无限', 'iph', '5c', '免费送', '红米', '购买权', '免费', '抽', '总有', '一款', '适合', '小伙', '速度', '来试', '运气', '人品', '爆发', ''], 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 23, 38, 48, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.67475, 16.52044], 'text': [' ', '爱', '王诗龄', 'kimi', '口', '水流', '干', '死', '翅', '可怜', ''], 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 22, 30, 16, tzinfo=datet
```

function:将data中的所有字典的text分割为词构成的列表

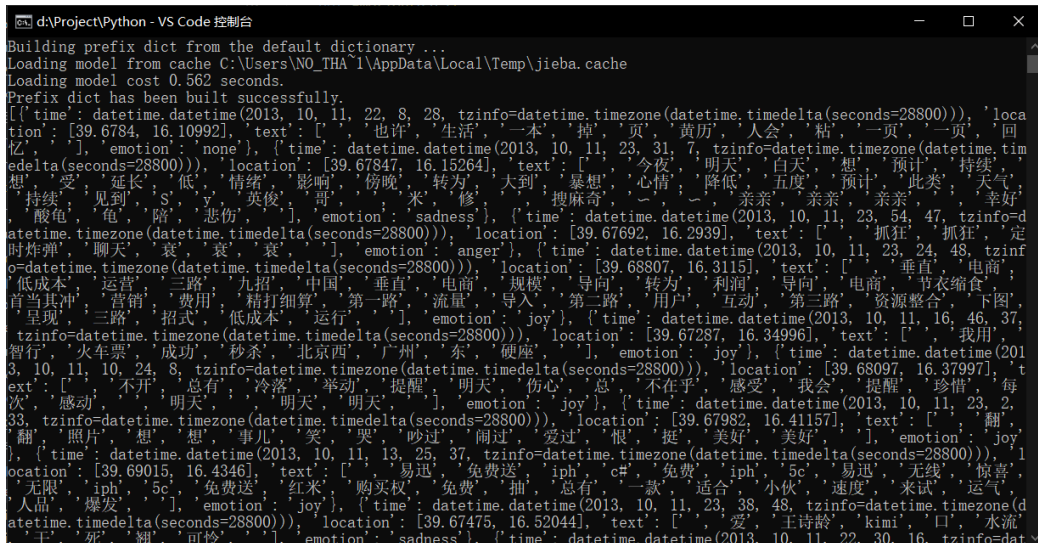
情绪分析闭包函数LoadEmotion()

```
1 def LoadEmotion(root):
2     '''
3     外函数加载情绪词
4     传入情绪词文件的根目录
5     返回情绪分析函数
6     '''
7     #载入词典
8     emotions={}
9     for root,dirs,files in os.walk(root):
10         for file in files:
11             emotion_name=file[:-4]
12             emotion_words={line.strip() for line in open(root+'\\'+file,'r')}
13             emotions.update({emotion_name:emotion_words})
14     def Emotion(data):
15         '''
16         内函数统计情绪
17         传入列表data(列表每个元素为time、location、text、emotion构成的字典)
18         返回修改过emotion的data列表
19         '''
20         for line in data:
21             nonlocal emotions#将载入的情绪词持久化
22             count_emotion={each:0 for each in emotions}
23             for word in line['text']:
24                 for emotion in emotions:
25                     if word in emotions[emotion]: count_emotion[emotion]+=1
26             line['emotion']=MaxEmotion(count_emotion)
27         return data
28     return Emotion
29
30 def MaxEmotion(dict):
31     '''
32     求字典最大值的键
33     处理多个最大值的情况
34     '''
```

```

35     max=-1
36     update=1
37     for each in dict:
38         if dict[each]<max: continue
39         if dict[each]>max: update=1
40         elif dict[each]==max: update=0
41         max_name=each
42         max=dict[each]
43
44     if update==0: return 'none'#有多个最大值
45     else: return max_name
46
47 def main():
48     '''
49     测试函数
50     '''
51     file_path='D:\\Project\\Python\\week3weibo\\weibo1.txt'#评论文件路径
52     dict_path='D:\\Project\\Python\\week3weibo\\stopwords_list.txt'#分词词典路径
53     emotion_lexicon_root='D:\\Project\\Python\\week3weibo\\emotion_lexicon'#情绪
54     data=ReadData(file_path)
55     data=CleanData(data)
56     data=CutText(data,dict_path)
57     Emotion=LoadEmotion(emotion_lexicon_root)
58     data=Emotion(data)
59     print(data)

```



```

d:\Project\Python - VS Code 控制台
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\NO_THA\1\AppData\Local\Temp\jieba.cache
Loading model cost 0.562 seconds.
Prefix dict has been built successfully.
[{'time': datetime.datetime(2013, 10, 11, 22, 8, 28, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.6784, 16.10992], 'text': [' ', '也许', '生活', '一本', '掉', '页', '黄页', '人会', '粘', '一页', '一页', '回忆', ''], 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 23, 31, 7, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.67847, 16.15264], 'text': [' ', '今夜', '明天', '白天', '想', '预计', '持续', '想', '受', '延长', '低', '情绪', '影响', '傍晚', '转为', '暴想', '心情', '降低', '五度', '预计', '此类', '天气', '持续', '见到', 'S', 'y', '英俊', '哥', '米', '修', '搜麻奇', '一', '一', '亲亲', '亲亲', '亲亲', '一', '幸好', '酸龟', '龟', '陪', '悲伤', ''], 'emotion': 'sadness'}, {'time': datetime.datetime(2013, 10, 11, 23, 54, 47, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.67692, 16.2939], 'text': [' ', '抓狂', '抓狂', '定时炸弹', '聊天', '衰', '衰', '衰', ''], 'emotion': 'anger'}, {'time': datetime.datetime(2013, 10, 11, 23, 24, 48, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.68807, 16.3115], 'text': [' ', '垂直', '电商', '低成本', '运营', '三路', '九招', '中国', '垂直', '电商', '规模', '导向', '转为', '利润', '导向', '电商', '节衣缩食', '首当其冲', '营销', '费用', '精打细算', '第一路', '流量', '导入', '第二路', '用户', '互动', '第三路', '资源整合', '下图', '呈现', '三路', '招式', '低成本', '运行', ''], 'emotion': 'joy'}, {'time': datetime.datetime(2013, 10, 11, 16, 46, 37, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.67287, 16.34996], 'text': [' ', '我用', '智行', '火车票', '成功', '秒杀', '北京西', '广州', '东', '硬座', ''], 'emotion': 'joy'}, {'time': datetime.datetime(2013, 10, 11, 23, 2, 33, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.67982, 16.41157], 'text': [' ', '翻', '翻', '照片', '想', '明天', '明天', ''], 'emotion': 'joy'}, {'time': datetime.datetime(2013, 10, 11, 23, 2, 33, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.67982, 16.41157], 'text': [' ', '翻', '翻', '照片', '想', '明天', '明天', ''], 'emotion': 'joy'}, {'time': datetime.datetime(2013, 10, 11, 13, 25, 37, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.69015, 16.4346], 'text': [' ', '易迅', '免费送', 'iph', 'c#', '免费', 'iph', '5c', '易迅', '无线', '惊喜', '无限', 'iph', '5c', '免费送', '红米', '购买权', '免费', '抽', '总有', '一款', '适合', '小伙', '速度', '来试', '运气', '人品', '爆发', ''], 'emotion': 'joy'}, {'time': datetime.datetime(2013, 10, 11, 23, 38, 48, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.67475, 16.52044], 'text': [' ', '爰', '王诗龄', 'kimi', '口', '水流', '干', '死', '翘', '可怜', ''], 'emotion': 'sadness'}, {'time': datetime.datetime(2013, 10, 11, 22, 30, 16, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.67475, 16.52044], 'text': [' ', '爰', '王诗龄', 'kimi', '口', '水流', '干', '死', '翘', '可怜', ''], 'emotion': 'sadness'}]

```

功能：通过闭包函数，只加载一次情绪词，计算data中每条评论的情绪

#认为每条评论只包含词频最大的一种情绪，且有多多个最大词频的情绪时，认为该评论没有情绪

时间模式函数TimeModel()

```

1 def TimeModel(data,model,emotion):
2     '''
3     计算时间模式
4     model是时间间隔秒数的整数
5     返回时间区间和频数的字典
6     '''
7     res={}
8     for line in tqdm(data):

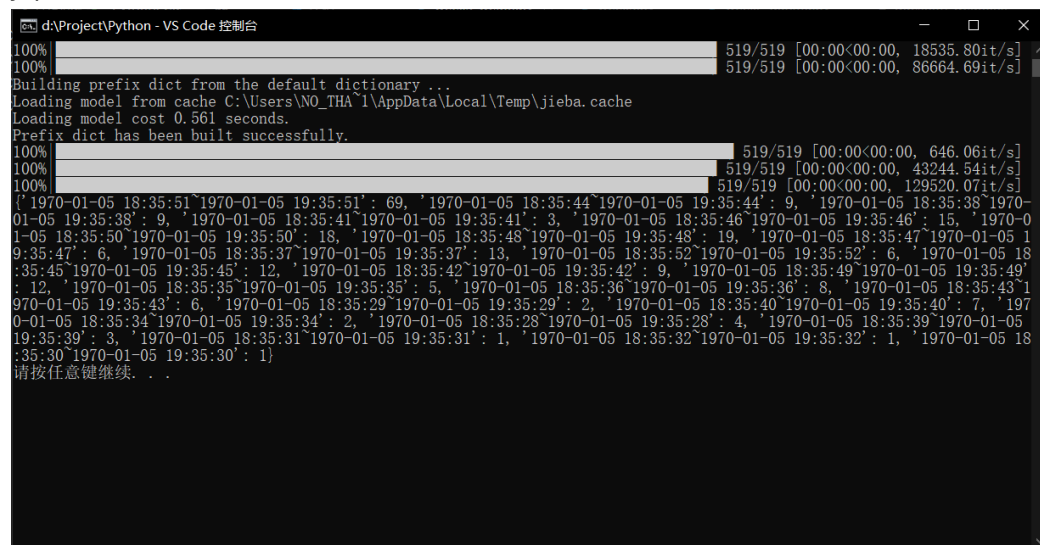
```

```

9         if line['emotion']!=emotion:#匹配情绪
10             continue
11         span=int(int(line['time'])/model)
12         t_struct=time.localtime(span)
13         start=str(time.strftime("%Y-%m-%d %H:%M:%S",t_struct))
14         span+=model
15         t_struct=time.localtime(span)
16         end=str(time.strftime("%Y-%m-%d %H:%M:%S",t_struct))
17         key_name=start+'~'+end
18         if key_name not in res:#加入新区间
19             res.update({key_name:1})
20         else:#已有区间
21             res[key_name]+=1
22     return res
23
24 def main():
25     '''
26     测试函数
27     '''
28     file_path='D:\\Project\\Python\\week3weibo\\weibo1.txt'#评论文件路径
29     dict_path='D:\\Project\\Python\\week3weibo\\stopwords_list.txt'#分词词典路径
30     emotion_lexicon_root='D:\\Project\\Python\\week3weibo\\emotion_lexicon'#情绪
31     data=ReadData(file_path)
32     data=CleanData(data)
33     data=CutText(data,dict_path)
34     Emotion=LoadEmotion(emotion_lexicon_root)
35     data=Emotion(data)
36     time_model=TimeModel(data,1*60*60,'anger')
37     print(time_model)
38

```

joy情绪的小时模式：



sadness情绪的周模式


```

d:\Project\Python - VS Code 控制台
100% 519/519 [00:00<00:00, 17892.85it/s]
100% 519/519 [00:00<00:00, 86595.74it/s]
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\NO_THA\1\AppData\Local\Temp\jieba.cache
Loading model cost 0.565 seconds.
Prefix dict has been built successfully.
100% 519/519 [00:00<00:00, 645.15it/s]
100% 519/519 [00:00<00:00, 43241.11it/s]
100% 519/519 [00:00<00:00, 173398.42it/s]
{'1970-01-01 08:38:04':1970-01-08 08:38:04': 240}
请按任意键继续. . .

```

(以前14946行数据为例)

功能：分析固定时间间隔内某一情绪的频数

空间分布函数DistanceRate()

```

1 def DistanceRate(data,centre,distance_standard,emotion):
2     '''
3     计算data中与centre欧式距离不大于distance的评论中,情绪为emotion的比例
4     '''
5     count_in=0
6     count_correct=0
7     for line in data:
8         dis=distance.euclidean(np.array(line['location']), np.array(centre))
9         if dis<=distance_standard:
10             count_in+=1
11             if line['emotion']==emotion:
12                 count_correct+=1
13     if count_in==0:
14         return -1.0
15     return count_correct/count_in
16
17 def main():
18     '''
19     测试函数
20     '''
21     file_path='D:\Project\Python\week3weibo\weibo1.txt'#评论文件路径
22     dict_path='D:\Project\Python\week3weibo\stopwords_list.txt'#分词词典路径
23     emotion_lexicon_root='D:\Project\Python\week3weibo\emotion_lexicon'#情绪
24     centre=[39.000000,116.000000]#中心位置
25     distance=1#半径
26     data=ReadData(file_path)
27     data=CleanData(data)
28     data=CutText(data,dict_path)
29     Emotion=LoadEmotion(emotion_lexicon_root)
30     data=Emotion(data)
31     rate=DistanceRate(data,centre,distance,'joy')
32     print(rate)

```

```

d:\Project\Python - VS Code 控制台
100% 519/519 [00:00<00:00, 18531.06it/s]
100% 519/519 [00:00<00:00, 74145.71it/s]
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\NO_THA\1\AppData\Local\Temp\jieba.cache
Loading model cost 0.560 seconds.
Prefix dict has been built successfully.
100% 519/519 [00:00<00:00, 639.38it/s]
100% 519/519 [00:00<00:00, 43181.92it/s]
0.44594594594594594
请按任意键继续. . .
```

功能：计算data中与centre欧式距离不大于distance的评论中,情绪为emotion的比例

总程序

```

1  import jieba
2  import re
3  import os
4  import time
5  from tqdm import tqdm
6  import numpy as np
7  from scipy.spatial import distance
8
9  def ReadData(file_path):
10     '''
11     数据读取函数
12     传入txt文件绝对路径
13     返回列表(列表每个元素为time、location、text、emotion构成的字典)
14     '''
15     f=open(file_path,'r',encoding='utf8')
16     data=[]
17     f.readline()#去表头
18     for line in tqdm(f.readlines()):
19         text=''
20         t=''
21         location=[]
22         #处理text
23         message=line.split()
24         if message[-2][0]!='+': continue#排除不含时间的错误数据
25         for each in message[2:-7]:
26             text+=' '+each
27         #处理location
28         if IsFloat(message[0][1:-2])==0:
29             continue
30         location.append(float(message[0][1:-2]))
31         location.append(float(message[1][:-2]))
32         #处理time
33         for each in message[-6:]:
34             t+=' '+each
```



```

35         tt=time.strptime(t,' %a %b %d %H:%M:%S %z %Y')
36         data.append({'time':time.mktime(tt),'location':location,'text':text,
37 f.close()
38         return data
39
40 def IsFloat(number):
41     '''
42     判断是否是浮点数
43     '''
44     for each in number:
45         if each not in {'1','2','3','4','5','6','7','8','9','0','.'}:
46             return 0
47     return 1
48
49 def CleanData(data):
50     '''
51     清洗数据，去除重复值和噪声(url
52     传入列表data(列表每个元素为time、location、text、emotion构成的字典)
53     返回清洗后的列表(text的值为词构成的列表)
54     '''
55     res=[]
56     url=re.compile(r'[http|https]*://[a-zA-Z0-9.?/&=:]*)'#取url
57     for each in tqdm(data):
58         each['text']=re.sub(url,'',each['text'])#去url
59         if each not in res:#去重
60             res.append(each)
61     return res
62
63 def CutText(data,dict_path):
64     '''
65     将data中的text分词
66     传入列表data(列表每个元素为time、location、text、emotion构成的字典)和词典地址
67     返回text分词后的列表
68     '''
69     f=open(dict_path,'r',encoding='utf8')
70     stop_words = [line.strip() for line in f.readlines()]
71     jieba.load_userdict(dict_path)#载入词典
72     for each in tqdm(data):
73         words=[]
74         for word in list(jieba.cut(each['text'])):
75             if word not in stop_words:#过滤停用词
76                 words.append(word)
77         each['text']=words
78     f.close()
79     return data
80
81 def LoadEmotion(root):
82     '''
83     外函数加载情绪词
84     传入情绪词文件的根目录
85     返回情绪分析函数

```

```

86     '''
87     #载入词典
88     emotions={}
89     for root,dirs,files in os.walk(root):
90         for file in files:
91             emotion_name=file[:-4]
92             emotion_words=[line.strip() for line in open(root+'\\'+file,'r')]
93             emotions.update({emotion_name:emotion_words})
94     def Emotion(data):
95         '''
96         内函数统计情绪(引用MaxEmotion函数)
97         传入列表data(列表每个元素为time、location、text、emotion构成的字典)
98         返回修改过emotion的data列表
99         '''
100         for line in tqdm(data):
101             nonlocal emotions#将载入的情绪词持久化
102             count_emotion={each:0 for each in emotions}
103             for word in line['text']:
104                 for emotion in emotions:
105                     if word in emotions[emotion]: count_emotion[emotion]+=1
106             line['emotion']=MaxEmotion(count_emotion)
107         return data
108     return Emotion
109
110 def MaxEmotion(dict):
111     '''
112     求字典最大值的键
113     处理多个最大值的情况
114     '''
115     max=-1
116     update=1
117     for each in dict:
118         if dict[each]<max: continue
119         if dict[each]>max: update=1
120         elif dict[each]==max: update=0
121         max_name=each
122         max=dict[each]
123     if update==0: return 'none'#有多个最大值
124     else: return max_name
125
126 def TimeModel(data,model,emotion):
127     '''
128     计算时间模式
129     model是时间间隔秒数的整数
130     返回时间区间和频数的字典
131     '''
132     res={}
133     for line in tqdm(data):
134         if line['emotion']!=emotion:#匹配情绪
135             continue
136         span=int(int(line['time'])/model)

```

```

137         t_struct=time.localtime(span)
138         start=str(time.strftime("%Y-%m-%d %H:%M:%S",t_struct))
139         span+=model
140         t_struct=time.localtime(span)
141         end=str(time.strftime("%Y-%m-%d %H:%M:%S",t_struct))
142         key_name=start+'~'+end
143         if key_name not in res:#加入新区间
144             res.update({key_name:1})
145         else:#已有区间
146             res[key_name]+=1
147     return res
148
149 def DistanceRate(data,centre,distance_standard,emotion):
150     '''
151     计算data中与centre欧式距离不大于distance的评论中,情绪为emotion的比例
152     '''
153     count_in=0
154     count_correct=0
155     for line in data:
156         dis=distance.euclidean(np.array(line['location']), np.array(centre))
157         if dis<=distance_standard:
158             count_in+=1
159             if line['emotion']==emotion:
160                 count_correct+=1
161     if count_in==0:
162         return -1.0
163     return count_correct/count_in
164
165 def main():
166     '''
167     测试函数
168     '''
169     file_path='D:\Project\Python\week3weibo\weibo.txt'#评论文件路径
170     dict_path='D:\Project\Python\week3weibo\stopwords_list.txt'#分词词典路径
171     emotion_lexicon_root='D:\Project\Python\week3weibo\emotion_lexicon'#情绪
172     model=1*7*24*60*60
173     centre=[39.000000,116.000000]#中心位置
174     distance=1#半径
175     data=ReadData(file_path)
176     data=CleanData(data)
177     data=CutText(data,dict_path)
178     Emotion=LoadEmotion(emotion_lexicon_root)
179     data=Emotion(data)
180     time_model=TimeModel(data,model,'sadness')
181     rate=DistanceRate(data,centre,distance,'joy')
182     print(time_model)
183     print('rate =',rate)
184
185 if __name__=='__main__':
186     main()

```

