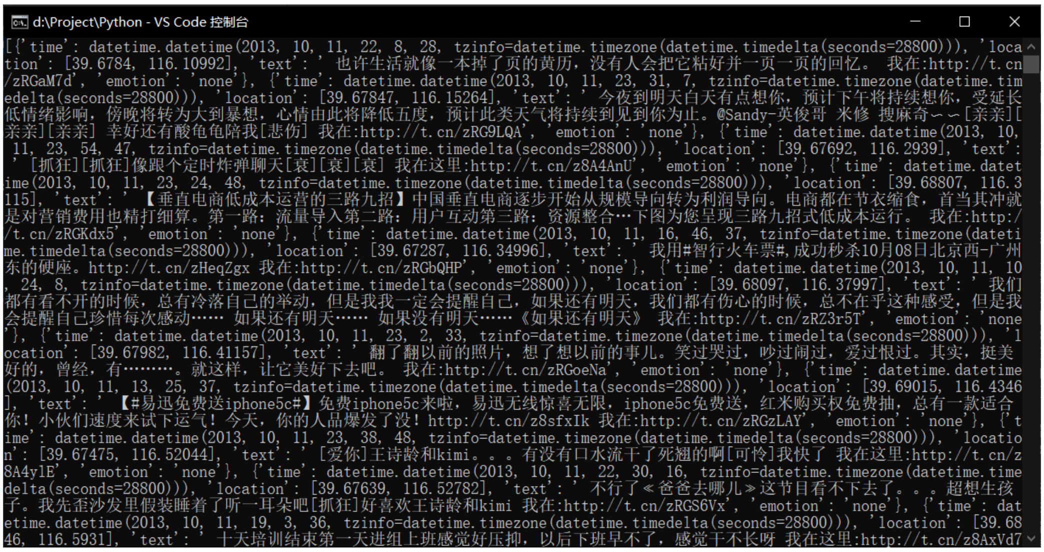
现代程序设计第3周作业

谢奕飞 20377077

数据读取函数ReadData()

```python
def ReadData(file_path):
    '''
    数据读取函数
    传入txt文件绝对路径
    返回列表(列表每个元素为time、location、text、emotion构成的字典)
    '''
    f=open(file_path,'r',encoding='utf8')
    data=[]
    f.readline()#去表头
    for line in f.readlines():
        text=''
        time=''
        location=[]
        #处理text
        message=line.split()
        for each in message[2:-7]:
            text+=' '+each
        #处理location
        location.append(float(message[0][1:-2]))
        location.append(float(message[1][1:-2]))
        #处理time
        for each in message[-6:]:
            time+=' '+each
        t=dt.datetime.strptime(time,' %a %b %d %H:%M:%S %z %Y')
        data.append({'time':t,'location':location,'text':text,'emotion':'none'})
    f.close()
    return data


def main():
    '''
    测试函数
    '''
    file_path='D:\Project\Python\week3weibo\weibo1.txt'
    data=ReadData(file_path)
    print(data)
```

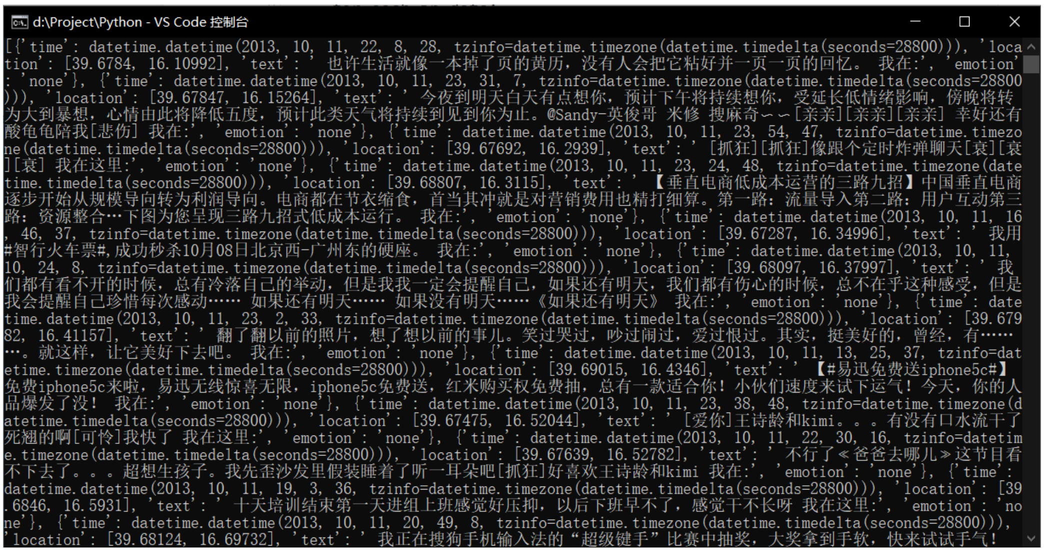


```
[{'time': datetime.datetime(2013, 10, 11, 22, 8, 28, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'loca
tion': [39.6784, 116.10992], 'text': ' 也许生活就像一本掉了页的黄历，没有人会把它粘好并一页一页的回忆。 我在:http://t.cn
/zRGaM7d', 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 23, 31, 7, tzinfo=datetime.timezone(datetime.tim
edelta(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][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.datet
ime(2013, 10, 11, 23, 24, 48, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.68807, 116.3
115], '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/zRGKdx5', 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 16, 46, 37, tzinfo=datetime.timezone(dateti
me.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))), 'l
ocation': [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, 13, 25, 37, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.69015, 116.4346
], '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'}, {'t
ime': datetime.datetime(2013, 10, 11, 23, 38, 48, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'locatio
n': [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/z
8A4ylE', 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 22, 30, 16, tzinfo=datetime.timezone(datetime.time
delta(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:http://t.cn/zRGS6Vx', 'emotion': 'none'}, {'time': dat
etime.datetime(2013, 10, 11, 19, 3, 36, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.68
46, 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()

```
def CleanData(data):
    '''
    清洗数据，去除重复值和噪声(url
    传入列表data(列表每个元素为time、location、text、emotion构成的字典)
    返回清洗后的列表(text的值为词构成的列表)
    '''
    res=[]
    url=re.compile(r'[http|https]*://[a-zA-Z0-9.?/&=:]*')#取url
    for each in data:
        each['text']=re.sub(url,'',each['text'])#去url
        if each not in res:#去重
            res.append(each)
    return res

def main():
    '''
    测试函数
    '''
    file_path='D:\Project\Python\week3weibo\weibo1.txt'
    data=ReadData(file_path)
    data=CleanData(data)
    print(data)
```

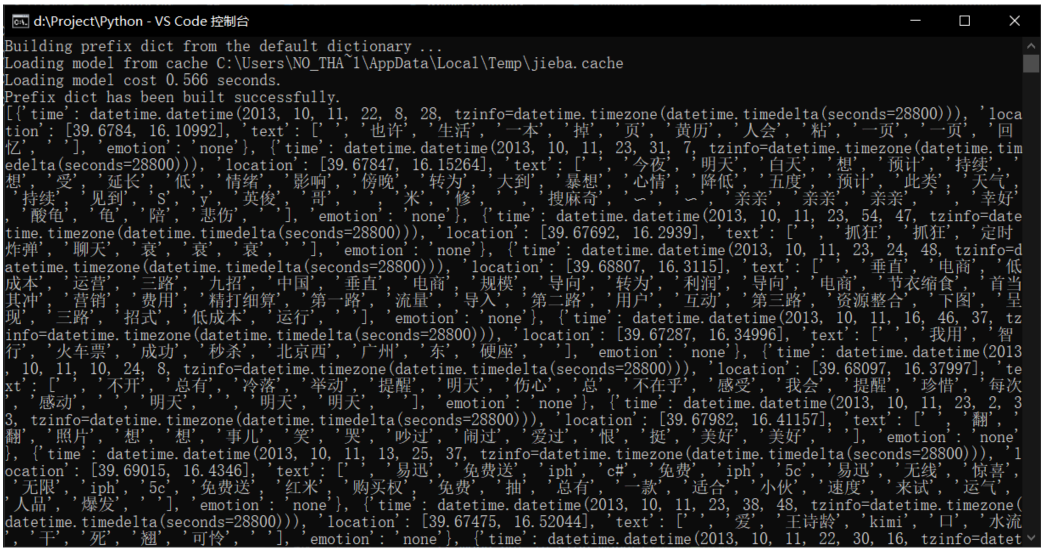


```
[{'time': datetime.datetime(2013, 10, 11, 22, 8, 28, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'loca
tion': [39.6784, 16.10992], 'text': ' 也许生活就像一本掉了页的黄历，没有人会把它粘好并一页一页的回忆。 我在:', 'emotion'
: 'none'}, {'time': datetime.datetime(2013, 10, 11, 23, 31, 7, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800
))), '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.timezo
ne(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(date
time.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': date
time.datetime(2013, 10, 11, 23, 2, 33, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.679
82, 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=dat
etime.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(d
atetime.timedelta(seconds=28800))), 'location': [39.67475, 16.52044], 'text': ' [爱你]王诗龄和kimi。。。有没有口水流干了
死翘的啊[可怜]我快了 我在这里:', 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 22, 30, 16, tzinfo=datetim
e.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': 'no
ne'}, {'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()

```
def CutText(data,dict_path):
    '''
    将data中的text分词
    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
    返回text分词后的列表
    '''
    f=open(dict_path,'r',encoding='utf8')
    stop_words = [line.strip() for line in f.readlines()]
    jieba.load_userdict(dict_path)#载入词典
    for each in data:
        words=[]
        for word in list(jieba.cut(each['text'])):
            if word not in stop_words:#过滤停用词
                words.append(word)
        each['text']=words
    f.close()
    return data

def main():
    '''
    测试函数
    '''
    file_path='D:\Project\Python\week3weibo\weibo1.txt'
    dict_path='D:\Project\Python\week3weibo\stopwords_list.txt'
    data=ReadData(file_path)
    clean_data=CleanData(data)
    seg_data=CutText(clean_data,dict_path)
    print(seg_data)
```

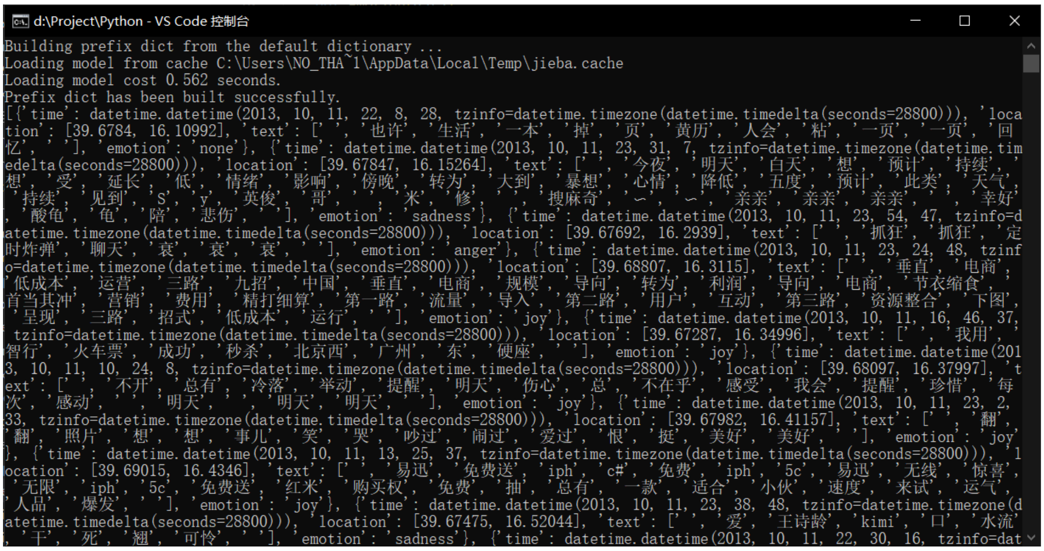


```
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
Loading model cost 0.566 seconds.
Prefix dict has been built successfully.
[{'time': datetime.datetime(2013, 10, 11, 22, 8, 28, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'loca
tion': [39.6784, 16.10992], 'text': [' ', '也许', '生活', '一本', '掉', '页', '黄历', '人会', '粘', '一页', '一页', '回
忆', ' '], 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 23, 31, 7, tzinfo=datetime.timezone(datetime.tim
edelta(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', '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', '∽', '∽', '亲亲', '亲亲', '亲亲', ' ', '幸好
', '酸龟', '龟', '陪', '悲伤', ' '], 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 23, 54, 47, tzinfo=date
time.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': 'none'}, {'time': datetime.datetime(2013, 10, 11, 23, 24, 48, tzinfo=d
atetime.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, tz
info=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], 'te
xt': [' ', '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', '明天', ' '], 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 23, 2, 3
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))), 'l
ocation': [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', '翘', '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()

```
def LoadEmotion(root):
    '''
    外函数加载情绪词
    传入情绪词文件的根目录
    返回情绪分析函数
    '''
    #载入词典
    emotions={}
    for root,dirs,files in os.walk(root):
        for file in files:
            emotion_name=file[:-4]
            emotion_words={line.strip() for line in open(root+'\\'+file,'r',
            emotions.update({emotion_name:emotion_words})
    def Emotion(data):
        '''
        内函数统计情绪
        传入列表data(列表每个元素为time、location、text、emotion构成的字典)
        返回修改过emotion的data列表
        '''
        for line in data:
            nonlocal emotions#将载入的情绪词持久化
            count_emotion={each:0 for each in emotions}
            for word in line['text']:
                for emotion in emotions:
                    if word in emotions[emotion]: count_emotion[emotion]+=1
            line['emotion']=MaxEmotion(count_emotion)
        return data
    return Emotion

def MaxEmotion(dict):
    '''
    求字典最大值的键
    处理多个最大值的情况
    '''
    max=-1
    update=1
    for each in dict:
        if dict[each]<max: continue
        if dict[each]>max: update=1
        elif dict[each]==max: update=0
        max_name=each
        max=dict[each]
    if update==0: return 'none'#有多个最大值
    else: return max_name

def main():
    '''
    测试函数
    '''
    file_path='D:\Project\Python\week3weibo\weibo1.txt'#评论文件路径
    dict_path='D:\Project\Python\week3weibo\stopwords_list.txt'#分词词典路径
    emotion_lexicon_root='D:\Project\Python\week3weibo\emotion_lexicon'#情绪
    data=ReadData(file_path)
    data=CleanData(data)
    data=CutText(data,dict_path)
    Emotion=LoadEmotion(emotion_lexicon_root)
    data=Emotion(data)
    print(data)
```



```
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))), 'loca
tion': [39.6784, 16.10992], 'text': [' ', '也许', '生活', '一本', '掉', '页', '黄历', '人会', '粘', '一页', '一页', '回
忆', ' '], 'emotion': 'none'}, {'time': datetime.datetime(2013, 10, 11, 23, 31, 7, tzinfo=datetime.timezone(datetime.tim
edelta(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', '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', '∽', '∽', '亲亲', '亲亲', '亲亲', ' ', '幸好'
, '酸龟', '龟', '陪', '悲伤', ' '], 'emotion': 'sadness'}, {'time': datetime.datetime(2013, 10, 11, 23, 54, 47, tzinfo=d
atetime.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, tzinf
o=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(201
3, 10, 11, 10, 24, 8, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'location': [39.68097, 16.37997], 't
ext': [' ', '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', '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', '想', '事儿', '笑', '哭', '吵过', '闹过', '爱过', '恨', '挺', '美好', '美好', ' '], 'emotion': 'joy'
}, {'time': datetime.datetime(2013, 10, 11, 13, 25, 37, tzinfo=datetime.timezone(datetime.timedelta(seconds=28800))), 'l
ocation': [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(d
atetime.timedelta(seconds=28800))), 'location': [39.67475, 16.52044], 'text': [' ', '爱', '王诗龄', 'kimi', '口', '水流'
, '干', '死', '翘', '可怜', ' '], 'emotion': 'sadness'}, {'time': datetime.datetime(2013, 10, 11, 22, 30, 16, tzinfo=dat
```

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

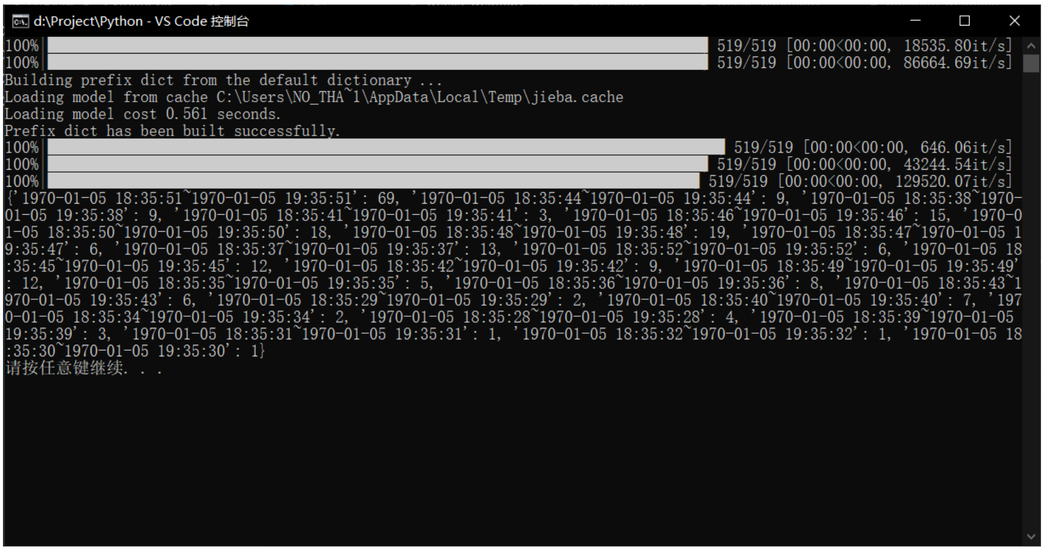
#认为每条评论只包含词频最大的一种情绪，且有多个最大词频的情绪时，认为该评论没有情绪

时间模式函数TimeModel()

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

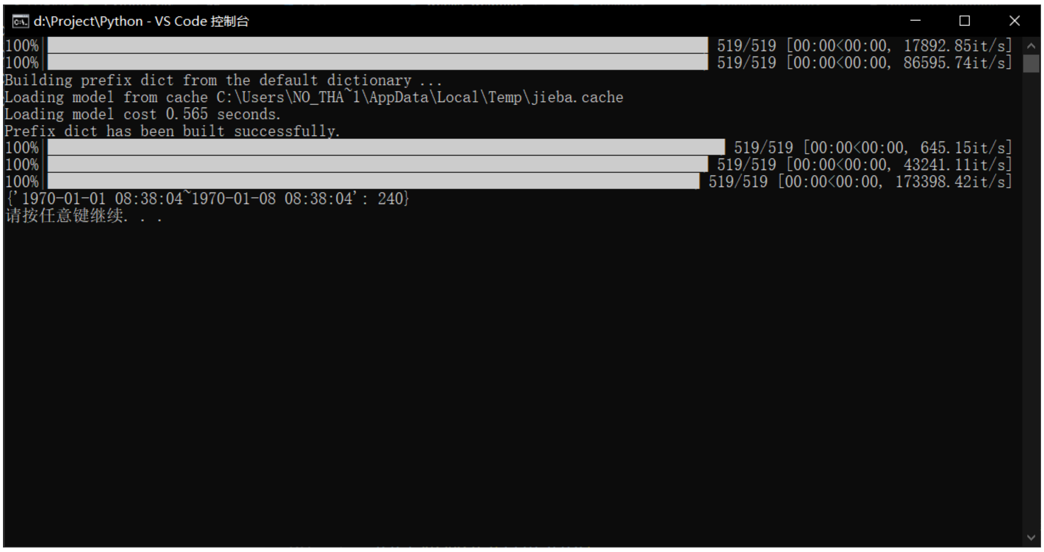
```python
        if line['emotion']!=emotion:#匹配情绪
            continue
        span=int(int(line['time'])/model)
        t_struct=time.localtime(span)
        start=str(time.strftime("%Y-%m-%d %H:%M:%S",t_struct))
        span+=model
        t_struct=time.localtime(span)
        end=str(time.strftime("%Y-%m-%d %H:%M:%S",t_struct))
        key_name=start+'~'+end
        if key_name not in res:#加入新区间
            res.update({key_name:1})
        else:#已有区间
            res[key_name]+=1
    return res

def main():
    '''
    测试函数
    '''
    file_path='D:\Project\Python\week3weibo\weibo1.txt'#评论文件路径
    dict_path='D:\Project\Python\week3weibo\stopwords_list.txt'#分词词典路径
    emotion_lexicon_root='D:\Project\Python\week3weibo\emotion_lexicon'#情绪
    data=ReadData(file_path)
    data=CleanData(data)
    data=CutText(data,dict_path)
    Emotion=LoadEmotion(emotion_lexicon_root)
    data=Emotion(data)
    time_model=TimeModel(data,1*60*60,'anger')
    print(time_model)

```

joy情绪的小时模式：



```
100%|██████████████████████████████████████████████| 519/519 [00:00<00:00, 18535.80it/s]
100%|██████████████████████████████████████████████| 519/519 [00:00<00:00, 86664.69it/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 0.561 seconds.
Prefix dict has been built successfully.
100%|███████████████████████████████████████████████| 519/519 [00:00<00:00, 646.06it/s]
100%|██████████████████████████████████████████████| 519/519 [00:00<00:00, 43244.54it/s]
100%|█████████████████████████████████████████████| 519/519 [00:00<00:00, 129520.07it/s]
{'1970-01-05 18:35:51~1970-01-05 19:35:51': 69, '1970-01-05 18:35:44~1970-01-05 19:35:44': 9, '1970-01-05 18:35:38~1970-
01-05 19:35:38': 9, '1970-01-05 18:35:41~1970-01-05 19:35:41': 3, '1970-01-05 18:35:46~1970-01-05 19:35:46': 15, '1970-0
1-05 18:35:50~1970-01-05 19:35:50': 18, '1970-01-05 18:35:48~1970-01-05 19:35:48': 19, '1970-01-05 18:35:47~1970-01-05 1
9:35:47': 6, '1970-01-05 18:35:37~1970-01-05 19:35:37': 13, '1970-01-05 18:35:52~1970-01-05 19:35:52': 6, '1970-01-05 18
:35:45~1970-01-05 19:35:45': 12, '1970-01-05 18:35:42~1970-01-05 19:35:42': 9, '1970-01-05 18:35:49~1970-01-05 19:35:49'
: 12, '1970-01-05 18:35:35~1970-01-05 19:35:35': 5, '1970-01-05 18:35:36~1970-01-05 19:35:36': 8, '1970-01-05 18:35:43~1
970-01-05 19:35:43': 6, '1970-01-05 18:35:29~1970-01-05 19:35:29': 2, '1970-01-05 18:35:40~1970-01-05 19:35:40': 7, '197
0-01-05 18:35:34~1970-01-05 19:35:34': 2, '1970-01-05 18:35:28~1970-01-05 19:35:28': 4, '1970-01-05 18:35:39~1970-01-05
19:35:39': 3, '1970-01-05 18:35:31~1970-01-05 19:35:31': 1, '1970-01-05 18:35:32~1970-01-05 19:35:32': 1, '1970-01-05 18
:35:30~1970-01-05 19:35:30': 1}
请按任意键继续. . .
```

sadness情绪的周模式



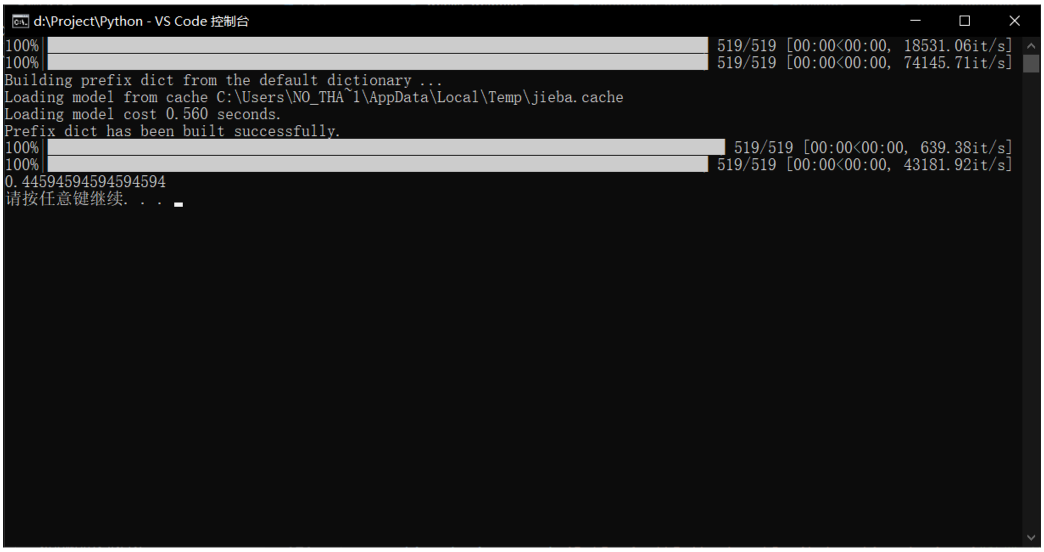
```
100%|████████████████████████████████████████| 519/519 [00:00<00:00, 17892.85it/s]
100%|████████████████████████████████████████| 519/519 [00:00<00:00, 86595.74it/s]
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\NO_THA~1\AppData\Local\Temp\jieba.cache
Loading 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()

```
def DistanceRate(data,centre,distance_standard,emotion):
    '''
    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
    '''
    count_in=0
    count_correct=0
    for line in data:
        dis=distance.euclidean(np.array(line['location']), np.array(centre))
        if dis<=distance_standard:
            count_in+=1
            if line['emotion']==emotion:
                count_correct+=1
    if count_in==0:
        return -1.0
    return count_correct/count_in

def main():
    '''
    测试函数
    '''
    file_path='D:\Project\Python\week3weibo\weibo1.txt'#评论文件路径
    dict_path='D:\Project\Python\week3weibo\stopwords_list.txt'#分词词典路径
    emotion_lexicon_root='D:\Project\Python\week3weibo\emotion_lexicon'#情绪
    centre=[39.000000,116.000000]#中心位置
    distance=1#半径
    data=ReadData(file_path)
    data=CleanData(data)
    data=CutText(data,dict_path)
    Emotion=LoadEmotion(emotion_lexicon_root)
    data=Emotion(data)
    rate=DistanceRate(data,centre,distance,'joy')
    print(rate)
```



```
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

```
import jieba
import re
import os
import time
from tqdm import tqdm
import numpy as np
from scipy.spatial import distance

def ReadData(file_path):
    '''
    数据读取函数
    传入txt文件绝对路径
    返回列表(列表每个元素为time、location、text、emotion构成的字典)
    '''
    f=open(file_path,'r',encoding='utf8')
    data=[]
    f.readline()#去表头
    for line in tqdm(f.readlines()):
        text=''
        t=''
        location=[]
        #处理text
        message=line.split()
        if message[-2][0]!='+': continue#排除不含时间的错误数据
        for each in message[2:-7]:
            text+=' '+each
        #处理location
        if IsFloat(message[0][1:-2])==0:
            continue
        location.append(float(message[0][1:-2]))
        location.append(float(message[1][:-2]))
        #处理time
        for each in message[-6:]:
            t+=' '+each
        tt=time.strptime(t,' %a %b %d %H:%M:%S %z %Y')
        data.append({'time':time.mktime(tt),'location':location,'text':text,
    f.close()
    return data

def IsFloat(number):
    '''
    判断是否是浮点数
    '''
    for each in number:
        if each not in {'1','2','3','4','5','6','7','8','9','0','.'}:
            return 0
    return 1

def CleanData(data):
    '''
    清洗数据，去除重复值和噪声(url
    传入列表data(列表每个元素为time、location、text、emotion构成的字典)
    返回清洗后的列表(text的值为词构成的列表)
    '''
    res=[]
    url=re.compile(r'[http|https]*://[a-zA-Z0-9.?/&=:]*')#取url
    for each in tqdm(data):
        each['text']=re.sub(url,'',each['text'])#去url
        if each not in res:#去重
            res.append(each)
    return res

def CutText(data,dict_path):
    '''
    将data中的text分词
    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
    返回text分词后的列表
    '''
    f=open(dict_path,'r',encoding='utf8')
    stop_words = [line.strip() for line in f.readlines()]
    jieba.load_userdict(dict_path)#载入词典
    for each in tqdm(data):
        words=[]
        for word in list(jieba.cut(each['text'])):
            if word not in stop_words:#过滤停用词
                words.append(word)
        each['text']=words
    f.close()
    return data

def LoadEmotion(root):
    '''
    外函数加载情绪词
    传入情绪词文件的根目录
    返回情绪分析函数
    '''
    #载入词典
    emotions={}
    for root,dirs,files in os.walk(root):
        for file in files:
            emotion_name=file[:-4]
            emotion_words={line.strip() for line in open(root+'\\'+file,'r',
            emotions.update({emotion_name:emotion_words})
    def Emotion(data):
        '''
        内函数统计情绪(引用MaxEmotion函数)
        传入列表data(列表每个元素为time、location、text、emotion构成的字典)
        返回修改过emotion的data列表
        '''
        for line in tqdm(data):
            nonlocal emotions#将载入的情绪词持久化
            count_emotion={each:0 for each in emotions}
            for word in line['text']:
                for emotion in emotions:
                    if word in emotions[emotion]: count_emotion[emotion]+=1
            line['emotion']=MaxEmotion(count_emotion)
        return data
    return Emotion

def MaxEmotion(dict):
    '''
    求字典最大值的键
    处理多个最大值的情况
    '''
    max=-1
    update=1
    for each in dict:
        if dict[each]<max: continue
        if dict[each]>max: update=1
        elif dict[each]==max: update=0
        max_name=each
        max=dict[each]
    if update==0: return 'none'#有多个最大值
    else: return max_name

def TimeModel(data,model,emotion):
    '''
    计算时间模式
    model是时间间隔秒数的整数
    返回时间区间和频数的字典
    '''
    res={}
    for line in tqdm(data):
        if line['emotion']!=emotion:#匹配情绪
            continue
        span=int(int(line['time'])/model)
        t_struct=time.localtime(span)
        start=str(time.strftime("%Y-%m-%d %H:%M:%S",t_struct))
        span+=model
        t_struct=time.localtime(span)
        end=str(time.strftime("%Y-%m-%d %H:%M:%S",t_struct))
        key_name=start+'~'+end
        if key_name not in res:#加入新区间
            res.update({key_name:1})
        else:#已有区间
            res[key_name]+=1
    return res

def DistanceRate(data,centre,distance_standard,emotion):
    '''
    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
    '''
    count_in=0
    count_correct=0
    for line in data:
        dis=distance.euclidean(np.array(line['location']), np.array(centre))
        if dis<=distance_standard:
            count_in+=1
            if line['emotion']==emotion:
                count_correct+=1
    if count_in==0:
        return -1.0
    return count_correct/count_in

def main():
    '''
    测试函数
    '''
    file_path='D:\Project\Python\week3weibo\weibo.txt'#评论文件路径
    dict_path='D:\Project\Python\week3weibo\stopwords_list.txt'#分词词典路径
    emotion_lexicon_root='D:\Project\Python\week3weibo\emotion_lexicon'#情绪
    model=1*7*24*60*60
    centre=[39.000000,116.000000]#中心位置
    distance=1#半径
    data=ReadData(file_path)
    data=CleanData(data)
    data=CutText(data,dict_path)
    Emotion=LoadEmotion(emotion_lexicon_root)
    data=Emotion(data)
    time_model=TimeModel(data,model,'sadness')
    rate=DistanceRate(data,centre,distance,'joy')
    print(time_model)
    print('rate =',rate)

if __name__=='__main__':
    main()
```