程设第一次作业

20377383 樊思涵

任务 1.

先将原始 csv 文件读入并检测输出结果

```
filename_0=r'C:\Users\LF\Desktop\弹幕数据\danmuku.csv'       #原始需读入文件
filename_1=r'C:\Users\LF\Desktop\弹幕数据\items.csv'         #需写入文件

data = []
text_all = ''
with open(filename_0,encoding='UTF-8') as csvfile:
    csv_reader = csv.reader(csvfile)    # 使用csv.reader读取csvfile中的文件
    header = next(csv_reader)           # 读取第一行每一列的标题
    for row in csv_reader:              # 将csv 文件中的数据保存到data中
        data.append(row[0])             # 选择弹幕列加入到data数组中
        text_all += row[0]              #连接所有弹幕字符串
##print(len(data))
##print(text)
```

测试结果如下：

```
7', '462981',
py_code> python -u "e:\code\py_code\week2\_mian.py"
 '4h', '四小时前！！！', '小程12点睡', '刚刚', '来辽 两小时前！', '2小时前！', '囍', '来了', '两小时前', '来了来了', '两小时！', '3小时
, '来了', '1小时前', '来了', '哇 来武汉啦 欢迎呀', '2小时前', '两小时前', '四小时', '来啦', '2小时', '哇塞 还好我吃了饭', '给年轻老板上
', '两小时前！', '三小时前', '点外卖准备', '来了', '2小时', '两小时前', '来啦来啦', '2小时前', '来了来了', '2小时前', '111', '来了来了
'2小时前', '4h前', '武汉吗？', '四小时前', '来啦！', '三年', '两小时前', '来辽！！！！', '三小时前', '两小时前～', '点了外卖等着', '两
看他俩的视频都好饿', '3小时前', '骄傲，看你们有广告高兴', '肯定是大佬配的音！！省钱了啊盗月社', '来报道了~', '两个小时前', '哇哇哇', '两
来了', '三小时前', '来了来了来了', '2个小时前', '两小时前', '3小时', '两个小时！', '三个小时前', '哈哈哈哈哈哈哈哈', '三小时前', '来啦
的有种人生一串的感觉了哈哈', '哈哈哈哈', '你们几小时前出生？', '吃饱了来看的', '4小时前', '更新了', '2小时前', '三个小时前', '2小时前',
刚看完浪浪回来', '2小时前！！！', '来啦', '？？？？', '3小时前', '好长', '刚下课', '社员报道', '3小时前', '三小时前热热还能吃', '两个小时
小时前', '三小时前', '看这两位视频极度舒适呀', '两小时前', '哇', '又是这个。。。', '来啦', '有广告啦！！！开心', '两小时', '时长感人',
念的？', '3小时前！', '我靠 刚从浪浪那里看了云南白药', '来了来了！！！！最近这个速度太爽了吧', '来了！', '两小时前', '两小时前', '4小时前
助了，恭喜恭喜', '看你们有广告高兴', ' 恰饭囉', '来啦', '看我刷到了什么', '怎么回事', '？？？', '大佬！！！！', '两小时前', '有赞助了！
台', '刚看完的视频也是这个牙膏广告', '终于有赞助了！', '？？？', '两小时前 耶耶耶', '哈哈哈武汉', '来了', '两小时前', '两小时前！！！',
？？？', '三小时前', '怎么到处都是这个恰饭商', '3小时前', '四个小时前', '哇哦', '啊啊啊啊', '来宜昌呗', '老公声音真好听', '终于等来',
, '开始吃饭lo', '4小时前啊', '竟然是云南白药', '大佬声音跪', '我去', '有赞助了呀', '有赞助了！', '都有赞助了', '我还以为我进了人生一串
'两小时前', '两小时前！', '一分钟前', '三小时', '恰饭', '4小时前', '四小时前', '云南白药牙膏对牙龈不好', '小程12点睡觉', '硬核？', '哈
, '？？', '可以可以光明正大', '两小时前', 'space门口吗', '大品牌', '三小时前', '要去武汉上大学啦啊啊啊啊', '我的家乡', '3小说', '终于有
卧槽我在武汉呀', '我刚看完小肛和阿灿的蜡像视频 这后面那个人哈哈哈哈哈哈哈', '啊啊啊啊啊啊你们在武汉啊！', '又是你云南白药！', '四个小时前
来武汉，，，', '哇哇哇 来武汉了！', '两小时', '小缸和阿灿的蜡像？', '武汉，捂汗，傻傻分不清楚', '哇塞哇塞哇塞', '啊啊啊啊啊啊啊我刚离开',
白药牙膏好评', '谐音梗，扣钱', '三小时前抢救抢救还能吃', '2小时前', '吉庆街吗？', '记笔记 回去吃', '前两个小时！', '武汉呀！！！！', '来
```

```
PS E:\code\py_code> python -u "e:\code\py_code\week2\_mian.py"
2799001
PS E:\code\py_code>
```

使用 jieba 库进行分词，并将词频统计后输出新的 csv 文件

```
#使用精确模式进行分词
count  = jieba.lcut(text_all)
##print(count)

#定义空字典，对分词结果进行词频统计
word_count={}
for word in count:
    word_count[word] = word_count.get(word, 0) + 1  #若字典存在key则频数加一否则创建key并令值为1

#按词频对分词进行排序
items = list(word_count.items())                    #把字典转换为列表方便排序
items.sort(key=lambda x: x[1], reverse=True)
print(items)

#输出个csv看一看
with open(filename_1, 'w',encoding='UTF-8',newline='') as f:
    writer = csv.writer(f)
    for i in items:
        writer.writerow(i)
```

结果如下：

	A	B	C
1	哈哈哈	1054303	
2	了	447265	
3	的	435470	
4	！	434143	
5	？	363231	
6	，	359492	
7	武汉	293070	
8		261264	
9	我	245360	
10	哈哈哈哈	219470	
11	啊	184209	
12	是	174830	
13	好	160194	
14	吃	148247	
15	。	134670	
16	加油	115976	
17	都	104409	
18	也	102846	
19	藕	93931	

任务 2.

在任务 1 中输出的表格可以发现停用词仍未被过滤，下面进行过滤停用词。

```
#添加停词表的新词频统计
word_count={}
stopwords = [line.strip() for line in open(filename_2,encoding='UTF-8').readlines()]
for word in count:
        if word not in stopwords and word != ' ':
            word_count[word] = word_count.get(word, 0) + 1
```

初次尝试发现分词表中的单空格字符串未能被正确添加，于是进行手动添加，新

的词频统计结果如下：

	A	B	C
1	哈哈哈哈	681	
2	武汉	537	
3	吃	411	
4	蒜	389	
5	藕	248	
6	好吃	228	
7	真的	196	
8	萝卜	181	
9	啊啊啊	136	
10	小时	135	
11	前	117	
12	恰饭	110	
13	热情	106	
14	大哥	98	
15	想	92	
16	买	89	
17	牛杂	86	
18	母上	85	
19	独头	83	
20	粉	82	
21	喜欢	75	

任务 3.

根据词频进行特征词筛选，只保留高频词，删除低频词出现次数少于 5 次的词，得到特征词组成的特征集。

```
#对items根据词频进行特征词筛选，只保留次数大于等于5的高频词
n_items=len(items)
##print(n_items)
items_new=[[i for i in range(2)]for i in range(n_items)]
j=0
for i in range(n_items):
    if items[i][1] >= 5:
        items_new[j][0] = items[i][0]
        items_new[j][1] = items[i][1]
    j+=1

#输出个csv看一看
with open(filename_1, 'w',encoding='UTF-8',newline='') as f:
    writer = csv.writer(f)
    for i in items_new:
        if i[0] != 0:
            writer.writerow(i)
```

结果如下：

603	倒	5
604	h	5
605	随便	5
606	缸	5
607	外地人	5
608	注册商标	5
609	大拇指	5
610	长堤	5
611	本地人	5
612	特产	5
613	绿	5
614	一头	5
615	开学	5
616	瞩目	5
617	令人	5
618	心疼	5
619	一种	5
620	香味	5
621	额	5
622	很帅	5
623	听到	5
624	汤包	5
625	矫口	5
626	孝感人	5
627	看不出来	5
628		

任务 4.

利用特征集为每一条弹幕生成向量表示，若弹幕含特征词则记录为‘1’否则为

‘0’。

下面抽取十条长度大于 15 的弹幕，试着用小弹幕集测试特征词是否能正确显示。

```python
#写个关键词列表函数
def flis_key(words):
    lis_key=[]
    for item in items_new:
        if item[0] == 0:
            break
        elif item[0] in words:
            lis_key.append(1)
        else:
            lis_key.append(0)
    return lis_key

#获得10条有效弹幕，并输出他们的关键词列表
lis_danmu=[]                                                  #一维列表，用来存储弹幕内容
lis_key_matrix=[]                                             #二维列表，用来存储10条弹幕的关键词0、1属性
for i in range(10):
    while True:
        number = random.randint(0,len(data)-1)                #获取随机弹幕的编号
        if len(data[number]) >= 15:                           #取字数大于15的有效弹幕
            break
    lis_danmu.append(data[number])
    lis_key_matrix.append(flis_key(data[number]))
    ##print(flis_key(data[number]))
print(lis_key_matrix[0])

#输出抽取的10条弹幕
print("抽取的10条弹幕分别是:")
print("\n".join(i for i in lis_danmu))
```

输出结果：

```
抽取的10条弹幕分别是:
最讨厌萝卜里的丝（太老了），极其影响口感
云南白药最近有点火啊，好几个美食up都接啦
个人觉得白萝卜能甩土豆十八条街
多给的话也不会多给太多的，少一点90也不亏
广东的萝卜牛杂里面的萝卜 巨好吃
母上也太富态了叭！！！哈哈哈哈哈哈哈
我家因为没种好种出来的全是独瓣蒜
前面说兰州的等等我！！兰州最近真的太凉快了
。。四川重庆方言是从湖北湖南传过去的
我巨爱吃藕，一次吃一根都没问题
```

其中第一条弹幕的特征集列表为：

```
[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

任务 5.

对特征集列表进行处理，找到‘1’数量最多的最典型弹幕

```
#找到随机10条弹幕中1数量最多的最典型弹幕
lis_total_key=[0 for i in range(len(lis_danmu))]
count_1=0
for lis in lis_key_matrix:
    for i in lis:
        lis_total_key[count_1]+=i
    count_1+=1
##print(lis_total_key)
n_typical = lis_total_key.index(max(lis_total_key))
print("最典型的弹幕是\"%s\"" %(lis_danmu[n_typical]))
```

计算每个弹幕之间的欧氏距离

```
#写个计算欧式距离的函数
def fdis(lis_a,lis_b):
    '''
    需要两个lis的长度相同
    '''
    s=0
    for i in range(len(lis_a)):
        if lis_a[i] != lis_b[i]:
            s+=1
    return math.sqrt(s)
##print(fdis(lis_key_matrix[0],lis_key_matrix[1]))

#计算每个弹幕距离其他弹幕的距离
lis_dis_matrix=[[0 for i in range(10)]for i in range(10)]
for i in range(10):
    for j in range(10-i):
        lis_dis_matrix[i][j] = fdis(lis_key_matrix[i],lis_key_matrix[j])
```

找到并输出距离最大和最小的两个弹幕

```
#找到最大距离值
lis_col_dis_max = []
lis_col_num_max = []
for i in lis_dis_matrix:                                    #先找到最大每行的最大值
    lis_col_dis_max.append(max(i))
    lis_col_num_max.append(i.index(max(i)))
dis_max_num = lis_col_dis_max.index(max(lis_col_dis_max))                   #找到全部的最大值
##print(lis_dis_matrix)
print("抽取的10条弹幕中距离最远的两条弹幕是:\n\"%s\"和\"%s\"\n它们之间的欧式距离为%.2f."%(lis_danmu[dis_max_num],lis_danmu[lis_col_num_max[dis_max_num]],max(lis_col_dis_max)))

#同理找到最小距离值
for i in lis_dis_matrix:                                #排除完全相同的弹幕
    for j in range(len(i)):
        if int(i[j]) == 0:
            i[j]+=10
lis_col_dis_min = []
lis_col_num_min = []
for i in lis_dis_matrix:                                    #先找到最大每行的最小值
    lis_col_dis_min.append(min(i))
    lis_col_num_min.append(i.index(min(i)))
dis_min_num = lis_col_dis_min.index(min(lis_col_dis_min))                   #找到全部的最小值
##print(lis_dis_matrix)
print("抽取的10条弹幕中距离最近的两条弹幕是:\n\"%s\"和\"%s\"\n它们之间的欧式距离为%.2f."%(lis_danmu[dis_min_num],lis_danmu[lis_col_num_min[dis_min_num]],min(lis_col_dis_min)))
```

结果展示：

```
抽取的10条弹幕分别是:
。。。户部巷还是别去了真的。。。
哈哈哈哈哈哈哈哈武汉人好可爱啊！！！
老板:我一锅只有一个鸭腿，平时都没人买
等我有钱了一定换一部能闻见味道的手机
独头蒜跟普通蒜的区别就是，处理起来方便。。
哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈
感觉沐上跟央视的一个主持人很像，叫宫岩
像南昌的水煮,看肯定没有南昌的辣
是白萝卜，武汉的萝卜真的特好吃
哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈
最典型的弹幕是"等我有钱了一定换一部能闻见味道的手机"
抽取的10条弹幕中距离最远的两条弹幕是:
"老板:我一锅只有一个鸭腿，平时都没人买"和"等我有钱了一定换一部能闻见味道的手机"
它们之间的欧式距离为4.00.
抽取的10条弹幕中距离最近的两条弹幕是:
"哈哈哈哈哈哈哈哈武汉人好可爱啊！！！ "和"哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈 "
它们之间的欧式距离为2.00.
```

任务 6.

使用 wordcloud 库

```
155   #高频词可视化
156   w=wordcloud.WordCloud(width=1000,font_path="C:\\Windows\\Fonts\\simfang.ttf",height=700)
157   w.generate(new_text)
158   w.to_file("core words.png")
```

结果展示：