现代程序设计第2周作业

谢奕飞 20377077

> 利用python数据结构（list, dict, set等）完成简单的文本分析任务。弹幕是现下视频网站，尤其是短视频网站提供的关键功能之一。以B站为例，其有着特殊的弹幕文化，且在视频的不同部分往往会有不同话题的弹幕：比如在视频开头会出 现“来啦”“x小时前”“第一!”;在up主暗示一键三连之后常常会出现“下次一定”或者“你币有 了”;和up主建立默契之后，观众可以判断视频是否有恰饭，往往在广告之前会出现“要素察 觉”“恰饭”“快跑”等等。因此，弹幕经常被作为测度用户（viewer）与视频作者（up主）之间交互行为的关键数据。
> 本次作业提供的数据来自B站某知名up主，已上传至课程资料的data目录下，数据格式说明如下。
> a. 弹幕文件：danmuku.csv，为2799000 rows × 3 columns，本次作业仅使用第一列，即弹幕的文本内容。
> b. 停用词表示例，stopwords_list.txt
> 请大家尝试完成以下数据分析任务：
> 1. 使用danmuku.csv，其中一个弹幕可以视为一个文档（document），读入文档并分词（可以使用jieba或pyltp）。
> 2. 过滤停用词（可用stopwords_list.txt，或自己进一步扩充）并统计词频，输出特定数目的高频词和低频词进行观察。建议将停用词提前加入到jieba等分词工具的自定义词典中，避免停用词未被正确分词。
> 3. 根据词频进行特征词筛选，如只保留高频词，删除低频词（出现次数少于5之类），并得到特征词组成的特征集。
> 4. 利用特征集为每一条弹幕生成向量表示，可以是0，1表示（one-hot，即该特征词在弹幕中是否出现）也可以是出现次数的表示（该特征词在弹幕中出现了多少次）。注意，可能出现一些过短的弹幕，建议直接过滤掉。
> 5. 利用该向量表示，随机找几条弹幕，计算不同弹幕间的语义相似度，可尝试多种方式，如欧几里得距离或者余弦相似度等，并观察距离小的样本对和距离大的样本对是否在语义上确实存在明显的差别。请思考，这种方法有无可能帮助我们找到最有代表性的弹幕？

代码

CaculateWordsFrequency函数

```
def CaculateWordsFrequency(file_name, stopfile_name):
    f = open(file_name)
    df = pd.read_csv(f, encoding='utf-8', usecols=[0])#读取文件
    comments = list(df['content'])
    stop_f = open(stopfile_name, 'r', encoding='utf-8')#导入分词表
    jieba.load_userdict(stopfile_name)
    stop_words = set()
    for line in stop_f:
        line = line.rstrip('\n')
        if len(line):
            stop_words.add(line)
    stop_f.close()
    counts = {}#分词+记录频率
    seg_list = []
    documents = []
```

```
    for content in comments:
        cur_seg_list = jieba.lcut(content)
        seg_list.append(cur_seg_list)
        documents.append(' '.join(cur_seg_list))
        for seg in cur_seg_list:
            if seg not in stop_words:
                counts[seg] = counts.get(seg, 0) + 1
    counts = dict(sorted(counts.items(), key=lambda dc: dc[1], reverse=True)
    print('TOP10: {}\n'.format(list(counts.keys())[:10]))
    dic_word = {'word':counts.keys(), 'frequency':counts.values()}
    pd.DataFrame(dic_word).to_csv('word_counts.csv')
    f.close()
    stop_f.close()
```

OutputWordCloud函数

```
def OutputWordCloud(counts, n):#绘制词云图
    wordpng = wordcloud.WordCloud(background_color='white', height=700, widt
    wordpng.generate(" ".join(list(counts.keys())[:n]))
    wordpng.to_file("result.png")
```

OutputMainWord函数

```
def OutputMainWord(counts):#打印关键词
    word_list = [key for key, value in counts.items() if value > 3]
    word_list = list(set(word_list))
    word_id = {}
    for i in range(len(word_list)):
        word_id[word_list[i]] = i
    return set(word_list), word_id

```

CaculateEuclid_distance函数

```
def CaculateEuclid_distance(v1, v2):#计算欧氏距离
    dis=distance.euclidean(np.array(v1), np.array(v2))
    return dis

```

CaculateCosine_distance函数

```
def CaculateCosine_distance(v1, v2):#计算余弦相似度
    if np.dot(np.array(v1), np.array(v2)) > 0:
        return distance.cosine(np.array(v1), np.array(v2))
    return MAXDIS

```

RandomComment函数

```
def RandomComment(seg_list):#随机选取弹幕
    id = rd.randint(0, len(seg_list) - 1)
```

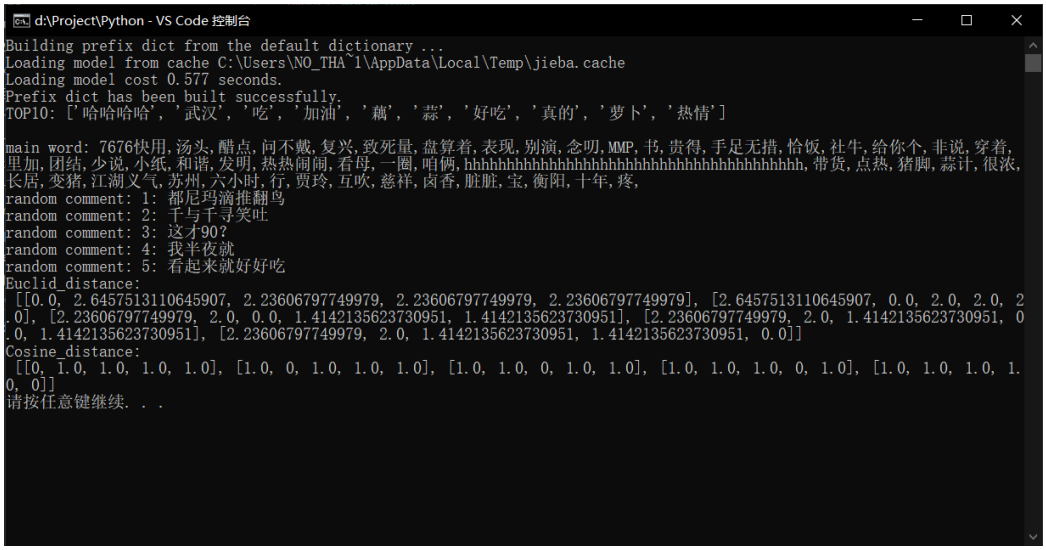
```
    random_vector = [0] * (len(main_word) + 5)
    for word in seg_list[id]:
        if word in main_word:
            random_vector[word_id[word]] = 1
    return comments[id], random_vector

```

主函数

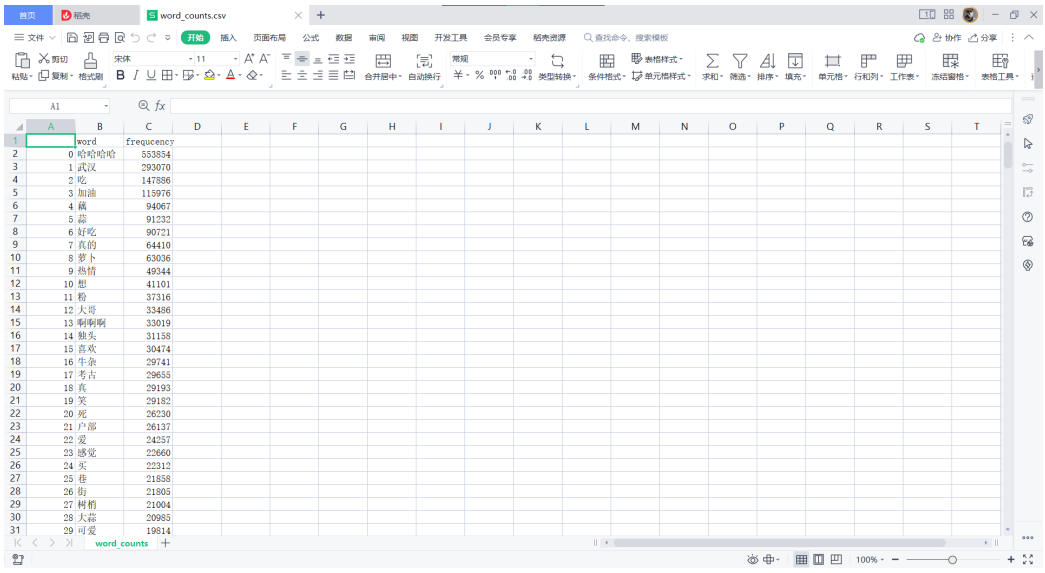
```
dic_words, counts, documents, seg_list, stop_words, comments = CaculateWords
OutputWordCloud(counts, 50)
main_word, word_id = OutputMainWord(counts)
print('词频特征词提取:', len(main_word), end='')
for i in range(50):
    print(rd.choice(list(main_word)), end=',')
print()
random_list = []
for i in range(5):
    content, vector = RandomComment(seg_list)
    random_list.append(vector)
    print('random comment: ', i+1, ': ', content, sep='')
euclid_dis = [[CaculateEuclid_distance(c1, c2) for c2 in random_list] for c1
cos_dis = [[CaculateEuclid_distance(c1, c2) for c2 in random_list] for c1 in
print('Euclid_distance:\n', euclid_dis)
print('Cosine_distance:\n', cos_dis)
```

运行结果



result.png

word_counts.csv



	A	B	C
1		word	frequency
2	0	哈哈哈	553854
3	1	武汉	293070
4	2	吃	147886
5	3	加油	115976
6	4	藕	94067
7	5	蒜	91232
8	6	好吃	90721
9	7	真的	64410
10	8	萝卜	63036
11	9	热情	49344
12	10	想	41101
13	11	粉	37316
14	12	大哥	33486
15	13	啊啊啊	33019
16	14	独头	31158
17	15	喜欢	30474
18	16	牛杂	29741
19	17	考古	29655
20	18	真	29193
21	19	笑	29182
22	20	死	26230
23	21	户部	26137
24	22	爱	24257
25	23	感觉	22660
26	24	买	22312
27	25	巷	21858
28	26	街	21805
29	27	树梢	21004
30	28	大蒜	20985
31	29	可爱	19814