第二周作业 20377019 詹嘉杰

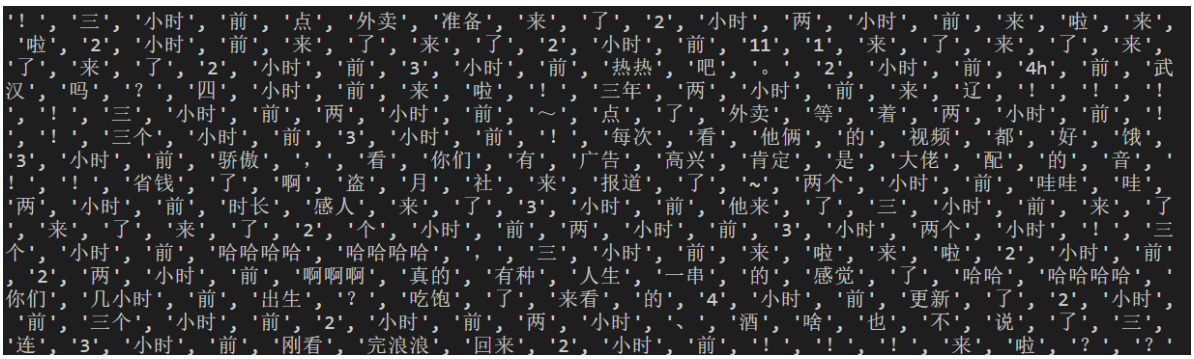
代码

```
from audioop import reverse
from ctypes import sizeof
from fileinput import filename
from pydoc import describe
from warnings import filterwarnings
import pandas as pd
import jieba
import collections
import wordcloud
import numpy as np
import jieba.analyse

#取出的弹幕数
dmnum = 100000

def read_csvfile(csvname):
    '''
    运用pandas读取csv文件中的content
    '''
    data = pd.read_csv(csvname)
    data1 = data['content']
    data1 = data1.head(dmnum)
    return data1

def filter_stopword(stwdname,words):
    '''
    读入停用词并筛选
    '''
    with open(stwdname,'r',encoding='utf-8') as file2:
        stop_words = file2.read()
    filtered_words = [word for word in words if word not in stop_words]
    return filtered_words

def split_word(data):
    '''
    分词
    '''
    jieba.load_userdict("stopwords_list.txt")
    wordlist = []
    for line in data:
        wordlist += jieba.lcut(str(line))
    return wordlist

def word_count(words):
    '''
    统计词频并删去词频小于5的词
    '''
    freq=collections.Counter(words)
    print("\n词频")
```

```
    print(freq)
    hifreqwords = {}
    for k in freq.keys():
        if (freq[k]>5):
            hifreqwords[k]=freq[k]
    #转换为字典按值排序
    hifreqwords = dict(sorted(hifreqwords.items(),key=lambda x:x[1],reverse =
True))
    print("\n高频词")
    print(hifreqwords)
    return freq,hifreqwords


def words_matrix(data,hifreqwords):
    '''
    通过高频词来为每条弹幕生成向量表示，并组成一个向量矩阵
    '''
    speset = list(hifreqwords.keys())
    length = len(speset)
    matrix = np.zeros((dmnum,length))
    for i in range(dmnum):
        spl = jieba.lcut(str(data1[i]))
        for j in range(length):
            if speset[j] in spl:
                matrix[i][j] = 1
            else:
                continue
    return matrix

def sen_dist(x,y,x1,y1):
    '''
    计算弹幕间的欧氏距离
    '''
    dist1 = np.sqrt(sum((x1-y1)**2))
    print("'" + str(x) + "'和'" + str(y) + "'"+"的距离是%.2f"%dist1)
    return

def tfidf(text):
    '''
    使用tfidf方法提取特征词
    '''
    tags = jieba.analyse.extract_tags(text,withWeight = True)
    worddic = {}
    for i in tags:
        worddic[i[0]]=i[1]
    print("\nTFIDF特征词")
    print(worddic)
    return worddic

def word_cloud(words):
    '''
    生成词云图
    '''
    wordtxt = " ".join(words)
```

```
    cloud =
wordcloud.WordCloud(font_path="STFANGSO.TTF",background_color='white',collocatio
ns=False,width=800,height=600,max_font_size=200)
    cloud.generate(wordtxt)
    cloud.to_file("词云图.jpg")


#读入数据
data1 = read_csvfile('danmuku.csv')

#导入自定义字典
jieba.load_userdict("stopwords_list.txt")

#分词
words = split_word(data1)
print(words)

#去除停用词
filtered_words = filter_stopword("stopwords_list.txt",words)

#统计词频
freq,hifreqwords = word_count(filtered_words)

matrix = words_matrix(data1,hifreqwords)

#选取了四条弹幕并计算他们之间的欧式距离
sen_dist(data1[5162],data1[5165],matrix[5162],matrix[5165])
sen_dist(data1[5151],data1[6817],matrix[5151],matrix[6817])

#TF-IDF提取关键词
text = ','.join(data1.tolist())
jieba.analyse.set_stop_words("stopwords_list.txt")
worddic = tfidf(text)

#生成词云
word_cloud(filtered_words)
```

1.读入数据并分词



2.过滤停用词后统计词频

由于处理时间过长就选取了前十万条弹幕

```
词频
Counter({'哈哈哈哈': 19143, '武汉': 6500, '吃': 5835, '蒜': 4482, '好吃': 3085, '藕': 2924, '真的': 2295, '热情': 1934, '萝卜': 1715, '恰饭': 1502, '啊啊啊': 1494, '粉': 1494, '想': 1441, '买': 1217, '大哥': 1210, '饿': 1161, '闻见': 1047, '户部': 920, '洛阳': 906, '独头': 883, '大蒜': 871, '江西': 865, '感觉': 858, '巷': 746, '便宜': 746, '爱': 741, '街': 720, '走': 662, '广告': 595, 'sao': 593, '牛杂': 589, '可爱': 580, '过期': 580, '云南白药': 550, '地方': 541, '声音': 532, '母上': 527, '牙膏': 520, '时间': 519, '湖北': 500, '味道': 499, '辣': 484, '警告': 482, '好像': 476, '嗦': 472, '树梢': 470, '卤': 459, '四川': 459, '哭': 454, '猝不及防': 449, '真实': 436, '闻到': 432, '恍恍惚惚': 431, '痔疮': 428, '肉': 420, '财大': 415, '长沙': 414, '好棒': 413, '广西': 407, '大佬': 402, '牛肉': 396, '买买': 391, '河南': 384, '卧槽': 377, '旁边': 368, '我家': 366, '好评': 361, '米粉': 361, '徐大': 352, '这条': 350, '莲藕': 349, '馋': 348, '岁': 342, '羡慕': 341, '去过': 340, '弹幕': 339, '商标': 339, '棒': 338, '摄像': 337, '这是': 333, '广东': 329, '找': 328, '烟': 327, '掉': 327, '湖南': 323, '手机': 322, '胖': 321, '小哥': 318, '南昌': 315, '沐': 315, '打钱': 314, '不吃': 313, '吉庆街': 308, '香': 307, '好好': 305, '老乡': 298, '氛围': 297, '排骨汤': 290, '脆': 290, '大叔': 289, '人来': 287, '不行': 277, '重庆': 277, '云南': 275, '诶': 270, '肛泰': 267, '长子': 264, '过年': 262, '老板': 253, '味': 250, '红红火火': 248, '围观': 245, '太好了': 245, '炒粉': 245, '我要': 239, '冬天': 238, '摄影': 237, '不到': 234, '汤': 234, '宝贝': 233, '下次': 228, '碗': 227, '马应龙': 227, '西苑': 226, '丑': 224, '云梦': 218, '米饭': 214, '玉子': 213, '卤菜': 211, '鸭': 211, '太棒了': 211, '有钱': 210, '口腔溃疡': 209, '厉害': 208, '解除': 208, '
```

3.将词频小于5的词删除

```
6, '人太多': 26, '直爽': 26, '特么': 26, '加碗': 26, '点汤': 26, '迟面': 26, '配蒜前': 26, '一两次': 26, '越往后': 26, '脖一绝': 26, '丁': 26, '天下无敌': 26, '很贵': 26, '格拉': 26, '欠': 26, '别拦': 26, '太惨': 26, '开会': 26, '一个月': 26, '次数': 26, '搓': 26, '一哈子': 26, '葱头': 26, '能生': 26, '手足无措': 26, '不赖': 26, '网友': 26, '才子': 26, '夏': 26, '东海': 26, '刘梅': 26, '视感': 26, '干哈': 26, '这腿': 26, '好瘦': 26, '景点': 26, '食': 26, '贫道': 26, '地核': 26, '汉服': 26, '之大': 26, '广场': 26, '两站': 26, '耿直': 26, '戴': 26, '墨镜': 26, '东方': 26, '卫生': 26, '人海战术': 26, '约等于': 26, '塑料盒': 26, '十倍': 26, '完美': 26, '牛大': 26, '超赞': 26, '起渣': 26, '发炎': 26, '格力': 26, '高救': 26, '一命': 26, '那年': 26, '太足': 26, '永远都是': 26, '热热闹闹': 26,
```

4.为弹幕生成向量表示

以前1000条弹幕为例：为每条弹幕创建一个向量，当特征词出现时为1，未出现为0，组成矩阵

```
高频词
{'小时': 114, '哈哈哈哈': 107, '武汉': 106, '蒜': 78, '吃': 30, '户部': 29, '啊啊啊': 26, '巷': 24, '大蒜': 24, '热情': 24, '好吃': 16, '吉庆街': 14, '旁边': 14, '母上': 12, '独头': 11, '老板': 10, '噜': 10, '真的': 9, '感觉': 9, '右边': 9, '云南白药': 8, '粮道': 8, '街': 8, '想': 8, '卤': 8, '四川': 8, '肯定': 7, '两个': 7, '赞助': 7, '来过': 7, '树梢': 7, '热干面': 7, '好像': 7, '藕': 7, '长子': 7, '沐': 7, '猜': 7, '视频': 6, '广告': 6, '终于': 6, '阿灿': 6, '北京': 6, '去过': 6, '瓶盖': 6, '辣': 6, '独蒜': 6}
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  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.]
 [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.]
 [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.]]
```

可以看到视频刚开始大家会发一些“x小时前”的消息，所以小时为最高频的词，而矩阵中第一列的1出现的行数也与“小时”出现在弹幕中的所在行一致。

5.向量表示计算欧式距离

```
'牛杂汤里面萝卜才是亮点啊'和'吃牛杂我也喜欢吃萝卜！'的距离是1.73
'牛腩煲里萝卜才是主角！！！'和'武汉吃烧烤，要蒜。老板说不知道什么是蒜'的距离是2.65
```

我选取了四条弹幕，分别是：

5163 牛杂汤里面萝卜才是亮点啊

5166 吃牛杂我也喜欢吃萝卜！

5152 牛腩煲里萝卜才是主角！！！

6818 武汉吃烧烤，要蒜。老板说不知道什么是蒜

而前两条的欧式距离小于后两条，则前两条的句意之间相似度比较大，而后两条的句意就完全不同。所以有可能选出最有代表性的弹幕

但这一方法还是依靠所包含的特征词差距来大概区分句意，无法对语意进行具体的分析，例如“喜欢吃萝卜”与“不喜欢吃萝卜”在这种判断方式中会被判断为句意一致。

6.词云图

7.TFIDF特征词对比

对前10000条弹幕处理

这是词频统计结果：

```
高频词
{'哈哈哈哈': 19143, '武汉': 6500, '吃': 5835, '蒜': 4482, '好吃': 3085, '藕': 2924, '真的': 2295, '热情': 1934
, '萝卜': 1715, '恰饭': 1502, '啊啊啊': 1494, '粉': 1494, '想': 1441, '买': 1217, '大哥': 1210, '饿': 1161, '
闻见': 1047, '户部': 920, '洛阳': 906, '独头': 883, '大蒜': 871, '江西': 865, '感觉': 858, '巷': 746, '便宜':
746, '爱': 741, '街': 720, '走': 662, '广告': 595, 'sao': 593, '牛杂': 589, '可爱': 580, '过期': 580, '云南白
药': 550, '地方': 541, '声音': 532, '母上': 527, '牙膏': 520, '时间': 519, '湖北': 500, '味道': 499, '辣': 484
, '警告': 482, '好像': 476, '嗦': 472, '树梢': 470, '卤': 459, '四川': 459, '哭': 454, '猝不及防': 449, '真实'
: 436, '闻到': 432, '恍恍惚惚': 431, '痔疮': 428, '肉': 420, '财大': 415, '长沙': 414, '好棒': 413, '广西': 40
7, '大佬': 402, '牛肉': 396, '买买': 391, '河南': 384, '卧槽': 377, '旁边': 368, '我家': 366, '好评': 361, '米
粉': 361, '徐大': 352, '这条': 350, '莲藕': 349, '馋': 348, '岁': 342, '羡慕': 341, '去过': 340, '弹幕': 339,
'商标': 339, '棒': 338, '摄像': 337, '这是': 333, '广东': 329, '找': 328, '烟': 327, '掉': 327, '湖南': 323, '
手机': 322, '胖': 321, '小哥': 318, '南昌': 315, '沐': 315, '打钱': 314, '不吃': 313, '吉庆街': 308, '香': 307
, '好好': 305, '老乡': 298, '氛围': 297, '排骨汤': 290, '脆': 290, '大叔': 289, '人来': 287, '不行': 277, '重
```

这是TFIDF特征词结果：

```
TFIDF特征词
{'哈哈哈哈': 1.1059683331467052, '武汉': 0.262259546612712, '好吃': 0.1488688008312836, '啊啊啊': 0.1139649735
6053413, '恰饭': 0.10699212749726382, '萝卜': 0.08870349012151574, '真的': 0.07704073617037378, '热情': 0.0757
1476945891233, '独头': 0.06289883394146735, '闻见': 0.06244130021291873, '大哥': 0.05049765720224204, '牛杂':
 0.04392032079856578, 'sao': 0.04224123276023798, '喜欢': 0.04199825572993887, '户部': 0.041783655271443045, '
大蒜': 0.040345281276477185, '母上': 0.037539847663820264, '江西': 0.03474609816900063, '洛阳': 0.034360299079
98212, '过期': 0.03383317362580292}
```

可以看出特征词的排名发生了较大的变化，“啊啊啊”的排名大幅提升，我认为是因为大家对于在弹幕中经常重复打出很多个“啊”，而且一般单独出现在一条弹幕中，不结合其他词语，使tfidf值提高。但总体出现的特征词依旧与高频词差不多，只是排序变化较大。

我认为TFIDF与词频相比，以词频统计的结果更好一些，因为弹幕普遍较短，且大家会经常重复同一句话在弹幕中，比如“哈哈哈”“啊啊啊”之类的，这时TFIDF会提高这些词的排名，但这些词并不能代表所有弹幕的内容，只能作为一个标准来区别具有真实内容的弹幕和多含重复的弹幕。