东北师范大学本科生课程论文

论文题目 《多媒体技术基础》课程论文

课程名称 多媒体技术基础

姓　　名 张瑞芳　　学　　号 2018013108

专　　业 数字媒体技术　　年　　级 2018 级

院 、 所 传媒科学学院　　年 月 日 2020.7.10

本科生课程论文评价标准

指 标	评价内容	评价等级（分值）				得分
		A	B	C	D	
选 题	选题是否新颖；是否有意义；是否与本门课程相关。	20-16	15-11	10-6	5-0	
论 证	思路是否清晰；逻辑是否严密；结构是否严谨；研究方法是否得当；论证是否充分。	20-16	15-11	10-6	5-0	
文 献	文献资料是否翔实；是否具有代表性。	20-16	15-11	10-6	5-0	
规 范	文字表达是否准确、流畅；体例是否规范；是否符合学术道德规范。	20-16	15-11	10-6	5-0	
能 力	是否运用了本门课程的有关理论知识;是否体现了科学研究能力。	20-16	15-11	10-6	5-0	
评阅教师签名： 年　月　日		总分：				

东北师范大学传媒科学学院（新闻学院）制

多媒体技术基础

1、研究内容

对改编自东野圭吾小说《白夜行》的电影《白夜行》的豆瓣短评进行情感分析；

2、数据采集

集搜客爬虫：
https://movie.douban.com/subject/4822829/comments?status=P

3、研究方法

朴素贝叶斯分类

4、Python 代码

```
import array
import re
from tkinter import _flatten

import matplotlib
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
from sklearn import metrics

from sklearn.metrics import classification_report
from sklearn.naive_bayes import MultinomialNB  #导入朴素贝叶斯分类器
from sklearn.model_selection import train_test_split #导入自动生成训练集和测试集的模块 train_test_split
import jieba
import wordcloud
from wordcloud import ImageColorGenerator, STOPWORDS

pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('expand_frame_repr', False)#禁止自动换行

f = pd.read_excel("白夜行 1.xlsx")
```

```
# 提取星级
f['star'] = f.rating_num.str.extract(r'(\d)')

# 删除列
f = f.drop(['rating_num', 'user_url', 'comment_time'], axis=1)
# 异常值
#f['content'] = f.content.replace('', '微笑')
print(f)

#数据预处理
#中文分词
contents = f['content']
contents = list(contents)
contents1 = list()
for content in contents:#去除标点符号
    comp = re.compile('[^A-Z^a-z^0-9^\u4e00-\u9fa5]')
    content = comp.sub('', content)
    contents1.append(content)
print("中文分词: ")
print(contents1)

#分词
stopwords = open("stopwords.txt", encoding="utf-8")
stopwords1 = list()
for stopword in stopwords.readlines():
    curLine = stopword.strip().split(" ")
    stopwords1.append(curLine)
stopwords1 = list(_flatten(stopwords1))#二维转一维
print("停用词: ")
print(stopwords1)

new_Series = pd.Series()
#处理停用词
new_list = list()
for content1 in contents1:
    ls = list(jieba.cut_for_search(content1))
    ls = [w for w in ls if w not in stopwords1]
    txt = " ".join(ls)
    new_list.append(txt)
print("去除停用词: ")
print(new_list)
new_Series = pd.Series(new_list)
f['content'] = new_Series
```

```
star = list(f['star'])
star1 = list()
for s in star:
    if s == s:
        star1.append(int(s))
    else:
        star1.append(0)

print("============================可视化
======================")
# 计数
star_num = f.star.value_counts()
star_num = star_num.sort_index()
print(star_num)
print("====================词云图====================")
print("===========绘制饼图================")
matplotlib.rcParams['font.family'] = 'Kaiti'#让中文字体正常显示
labels = '1 星', '2 星', '3 星', '4 星', '5 星'
sizes = list(star_num)
plt.pie(sizes, explode=None, labels=labels, autopct='%1.1f%%')
plt.title("豆瓣评分比例")
plt.axis('equal')
plt.show()
print("==================================三个词云图
========================")
text1 = f[(f.star=='4')|(f.star=='5')]['content']
positive = list(text1)
file = open("positive.txt", 'a')
for i in range(len(positive)):
    s = str(positive[i]).replace('[','').replace(']','')#去除[],
这两行按数据不同，可以选择
    s = s.replace("'",'').replace(',','') +'\n'   #去除单引号，逗
号，每行末尾追加换行符
    file.write(s)
alice_coloring = np.array(Image.open('people-flower.jpg'))
f_w = open("positive.txt", "r")
t = f_w.read()
ls_w = jieba.lcut(t)
txt_w = " ".join(ls_w)
w = wordcloud.WordCloud(font_path="msyh.ttc",
background_color="black",
                        mask=alice_coloring, collocations=False,
                        stopwords=['一点', '这部','片子','小时','半
```

```
小时',
                                '好好','电视剧','人物','之间','
电视']
                        )
w.generate(t)
image_color = ImageColorGenerator(alice_coloring)
plt.imshow(w, interpolation='bilinear')
plt.title("正向评分原因")
plt.axis('off')
plt.show()
w.to_file("positive.png")
print("==================负向评分原因===============")
text2 = f[(f.star=='1')|(f.star=='2')]['content']
negative = list(text2)
file = open("negative.txt", 'a')
for i in range(len(negative)):
    s = str(negative[i]).replace('[','').replace(']','')#去除[],
这两行按数据不同，可以选择
    s = s.replace("'",'').replace(',','') +'\n'   #去除单引号，逗
号，每行末尾追加换行符
    file.write(s)
alice_coloring = np.array(Image.open('people-flower.jpg'))
f_w = open("negative.txt", "r")
t = f_w.read()
ls_w = jieba.lcut(t)
txt_w = " ".join(ls_w)
w = wordcloud.WordCloud(font_path="msyh.ttc",
background_color="black",
                        mask=alice_coloring, collocations=False,
                        stopwords=['一点', '这部','片子','小时','半
小时',
                                '好好','电视剧','人物','之间','
小说','看过','电视']
                        )
w.generate(t)
image_color = ImageColorGenerator(alice_coloring)
plt.imshow(w, interpolation='bilinear')
plt.title("负向评分原因")
plt.axis('off')
plt.show()
w.to_file("negative.png")
print("============================中评原因
====================")
text3 = f[(f.star=='3')]['content']
```

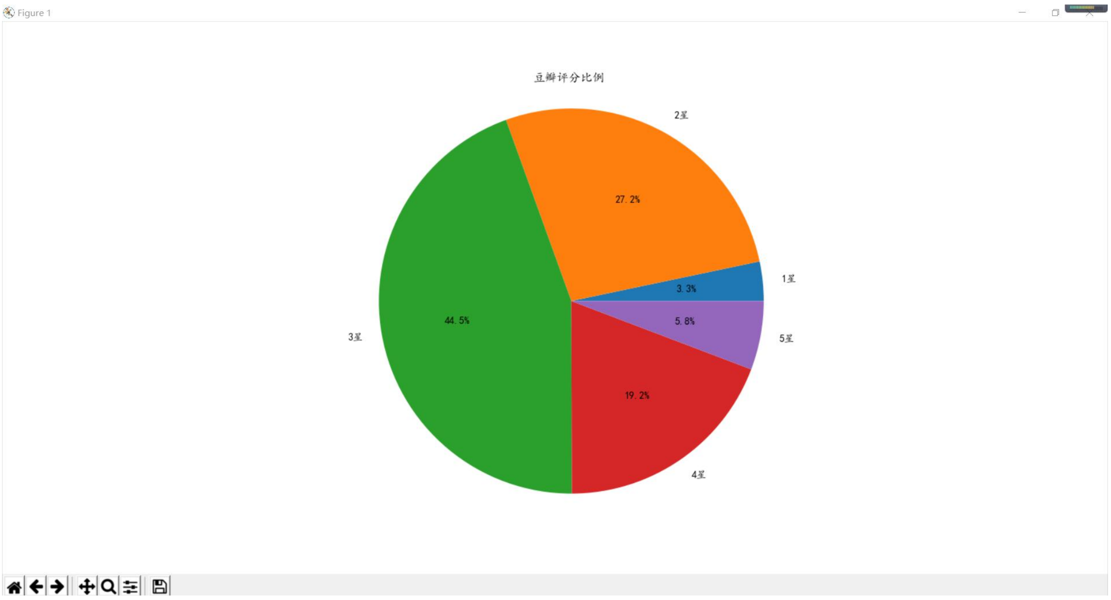
```
medium = list(text3)
file = open("medium.txt", 'a')
for i in range(len(medium)):
    s = str(medium[i]).replace('[','').replace(']','')#去除[],这
两行按数据不同，可以选择
    s = s.replace("'",'').replace(',','') +'\n'   #去除单引号，逗
号，每行末尾追加换行符
    file.write(s)
alice_coloring = np.array(Image.open('people-flower.jpg'))
f_w = open("medium.txt", "r")
t = f_w.read()
ls_w = jieba.lcut(t)
txt_w = " ".join(ls_w)
w = wordcloud.WordCloud(font_path="msyh.ttc",
background_color="black",
                        mask=alice_coloring, collocations=False,
                        stopwords=['一点', '这部','片子','小时','半
小时',
                                   '好好','电视剧','人物','之间','
小说','看过',
                                   '电视','看过','故事']
                        )
w.generate(t)
image_color = ImageColorGenerator(alice_coloring)
plt.imshow(w, interpolation='bilinear')
plt.title("中评评分原因")
plt.axis('off')
plt.show()
w.to_file("medium.png")
print("============================================================
============================")
star2 = list()
for st in star1:
    if st > 3:
        st = 1
        star2.append(st)
    else:
        st = 0
        star2.append(st)
f['star'] = star2
print(f['star'])
print(f)

print("============================朴素贝叶斯
```

```
======================")
clf = MultinomialNB()
x = f['content']
x = list(x)
y = f['star']
y = list(y)
n = len(x)//8
x_train, y_train = x[n:], y[n:]
x_train = pd.Series(x_train)
y_train = pd.Series(y_train)
x_test, y_test = x[:n], y[:n]
x_test = pd.Series(x_test)
y_test = pd.Series(y_test)


from sklearn.feature_extraction.text import CountVectorizer,
TfidfVectorizer
from sklearn.pipeline import Pipeline
from sklearn.naive_bayes import MultinomialNB
count_vec = CountVectorizer(max_df=0.8, min_df=3)
tfidf_vec = TfidfVectorizer()

def MNB_Classifier():
    return Pipeline([
        ('count_vec', CountVectorizer()),
        ('mnb', MultinomialNB())
    ])
mnbc_clf = MNB_Classifier()
# 进行训练
print("Start training...")
mnbc_clf.fit(x_train, y_train)
print("training done!")
answer_b = mnbc_clf.predict(x_test)
print("0：差评和中评；1：好评")
print(answer_b)
print("Prediction done!")
#准确率测试
accuracy=metrics.accuracy_score(y_test,answer_b)
print('准确率：'+str(accuracy))
print("The classification report for b:")
print(classification_report(y_test, answer_b))
```

5、实验运行结果

```
=============================朴素贝叶斯======================
Start training...
training done!
0：差评和中评；1：好评
[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]
Prediction done!
准确率：0.7903225806451613
```

6、可视化展示



（正向评分原因）


Figure 1
负向评分原因
感觉
呕
演
剧情
改编
故事
导演
真希
堀北
东野
原著
演技
细节
版本
选角
雪穗
真的

（负向评分原因）


Figure 1
中评评分原因
感觉
改编
版本
角色
电影
堀北
东野
原著
电影
剧情
日剧
两个
真的
雪穗

（中评原因）

6、结果分析

从饼图来看，中评占很大一部分；从正向评分和中评评分分析来看，有对原著情感的很大一部分原因在其中。

利用朴素贝叶斯进行情感分析，可以看出差评和中评要远大于好评，

```
The classification report for b:
              precision    recall  f1-score   support

           0       0.79      1.00      0.88        48
           1       1.00      0.07      0.13        14

    accuracy                           0.79        62
   macro avg       0.89      0.54      0.51        62
weighted avg       0.84      0.79      0.71        62
```

使用 classification_report 函数对分类结果，从精确率 precision、召回率 recall、f1 值 f1-score 和支持度 support 四个维度进行衡量，可以看到分类器的效果是不错的。也就是从测试集可以大概看出，电影《白夜行》改编并不好。