

实验三 分词和情感分析

【实验目的】

1. 了解分词的基本方法和原理。
2. 掌握情感分析的基本原理和实现方法。

【实验学时】

建议 2 学时

【实验环境配置】

- 1、Windows 环境
- 2、Anaconda
- 3、Jieba
- 4、Snownlp
- 5、Bosonnlp

【实验原理】

1、结巴工具集

参考链接：

- 1、<https://github.com/fxsjy/jieba>
- 2、<https://zhuanlan.zhihu.com/p/64409753>

2、snownlp 工具集

参考链接：

- 1、<https://github.com/isnowfy/snownlp>
- 2、<https://www.cnblogs.com/zhuminghui/p/10953717.html>

3、cnsenti 工具集

参考链接：

- 1、<https://github.com/thunderhit/cnsenti>

2、<https://zhuanlan.zhihu.com/p/117673231>

4、bosonnlp 工具集

参考链接：

- 1、<http://docs.bosonnlp.com/sentiment.html>
- 2、<https://tool.oschina.net/uploads/apidocs/jquery/regexp.html>

5、bosonnlp 的情感评分库。

参考链接：

- 1、http://static.bosonnlp.com/resources/BosonNLP_sentiment_score.zip

【实验步骤】

1、从相关网页抓取评价数据（通过 chrome 浏览器的 scraper 插件）。

● 使用 pycharm 抓取豆瓣电影《尚气与十环传奇》短评



因为豆瓣对抓取评论有页数限制，所以先登录然后在 network 中获取 cookies，并添加在程序中的 requests 的 cookies 字段中。

```
1. import csv
2. import requests
3. from bs4 import BeautifulSoup
4.
5.
6. def crawl_data(num):
```

```

7.     url = 'https://movie.douban.com/subject/30394797/comments?start=' +
        str(num) + '&limit=20&status=P&sort=new_score'
8.     header = {
9.         'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) '
10.        'Chrome/83.0.4103.116 Safari/537.36'}
11.     cookie_str = 'll="108099"; bid=UsY6fn6S2q0; _pk_ref.100001.4cf6=[
        "", "", 1637999102, "https://www.bing.com/"]; ' \
12.        '_pk_ses.100001.4cf6=*; __utma=30149280.1033624158.1
        637999102.1637999102.1637999102.1; ' \
13.        '__utmz=30149280.1637999102.1.1.utmcsr=bing|utmccn=(
        organic)|utmcmd=organic|utmctr=(not provided); ' \
14.        '__utmc=30149280; __utmb=223695111.0.10.1637999102;
        ' \
15.        '__utma=223695111.1120034313.1637999102.1637999102.1
        637999102.1; ' \
16.        '__utmz=223695111.1637999102.1.1.utmcsr=bing|utmccn=
        (organic)|utmcmd=organic|utmctr=(not provided); ' \
17.        '__utmc=223695111; ap_v=0,6.0; ' \
18.        '__gads=ID=7b7f8f7b7eef1c89-22a0da0747cf00b5:T=16379
        99103:RT=1637999103:S=ALNI_MYrFVvcuvUp' \
19.        '-LwU0D1FJxCKRZ1MRw; _vwo_uuid_v2=D3566E15C5EC5BA005
        701862452CFB6CA|b6e1b9cf0eb068b4e6189aea5c5e92b6' \
20.        '; trc_cookie_storage=taboola%20global%3Auser-id=71d
        b4c71-f0d0-4134-a1fc-0e60f005c4bb-tuct89b679c; ' \
21.        '__cc_id=308be1d9d8935fdab2029c51e871e9bb; panoramaId
        _expiry=1638604557948; ' \
22.        'panoramaId=d5bb6c986d4e3e52ed7dd22984c116d53938cfc9
        cd93631bab4883fc45458f88; ' \
23.        'dbcl2="250780697:ZloPULFBAYs"; ck=HtZI; push_noty_n
        um=0; push_doumail_num=0; __utmt=1; ' \
24.        '__utmv=30149280.25078; __utmb=30149280.2.10.1637999
        102; ' \
25.        '__yadk_uid=pvf21wlNJw4w1f03LRifplil2pmgsKti; ' \
26.        '_pk_id.100001.4cf6=049f7200c963d88f.1637999102.1.16
        38000110.1637999102. '
27.     cookies = {}
28.     # 处理 cookies
29.     for line in cookie_str.split(';'):
30.         name, value = line.strip().split('=', 1)
31.         cookies[name] = value
32.     # url = 'https://movie.douban.com/subject/30394797/comments?statu
        s=P'

```

```

33.     # all_url = 'https://movie.douban.com/subject/30394797/comments?s
    tart=20&limit=20&status=P&sort=new_score'
34.     try:
35.         res = requests.get(url, headers=header, cookies=cookies)
36.         return res.text
37.     except Exception as e:
38.         # print(e)
39.         print('错误')
40.
41.
42. def get_info(text):
43.     bs = BeautifulSoup(text, 'lxml')
44.     con_list = bs.select('.comment-item')
45.     review = {}
46.     for con in con_list:
47.         if (con.select('span.rating')):
48.             review['score'] = (con.select('span.rating')[0].get('titl
e'))
49.         else:
50.             review['score'] = '未知'
51.             review['content'] = con.select('span.short')[0].text.strip()
52.
53.             write_to_file(review)
54.         # for i in con_list:
55.         #     print(i.text)
56.
57. # 将字典形式的内容写入文件
58. def write_to_file(item):
59.     with open('review.csv', 'a', encoding='utf_8_sig', newline='') as
        f:
60.         fieldnames = ['content', 'score'] # 内容和分数
61.         w = csv.DictWriter(f, fieldnames=fieldnames)
62.         w.writerow(item)
63.
64.
65. def main():
66.     for i in range(0, 20):
67.         get_info(crawl_data(i * 20))
68.         print('-----爬取第%d 页完成
        -----' % (i + 1))
69.         # get_info(crawl_data(10))
70.
71.

```

```

72. if __name__ == '__main__':
73.     main()

```

评论内容字段“.comment-item”。

```

▶<div class="comment">...</div>
</div>
▼<div class="comment-item " data-cid="3024489759">
  ::before
  ▶<div class="avatar">...</div> == $0
  ▼<div class="comment">
    ▶<h3>...</h3>
    ▼<p class="comment-content">
      <span class="short">在美国的电影院，看一部美国主流电影，听着大段大段的
      边的美国人却要字幕，真的是很神奇的经验。</span>
    </p>
  </div>
</div>
▶<div class="comment-item " data-cid="3037933874">...</div>
▶<div class="comment-item " data-cid="3023261398">...</div>
▶<div class="comment-item " data-cid="3023579515">...</div>

```

用户评分字段“span.rating”中的“title”字段。

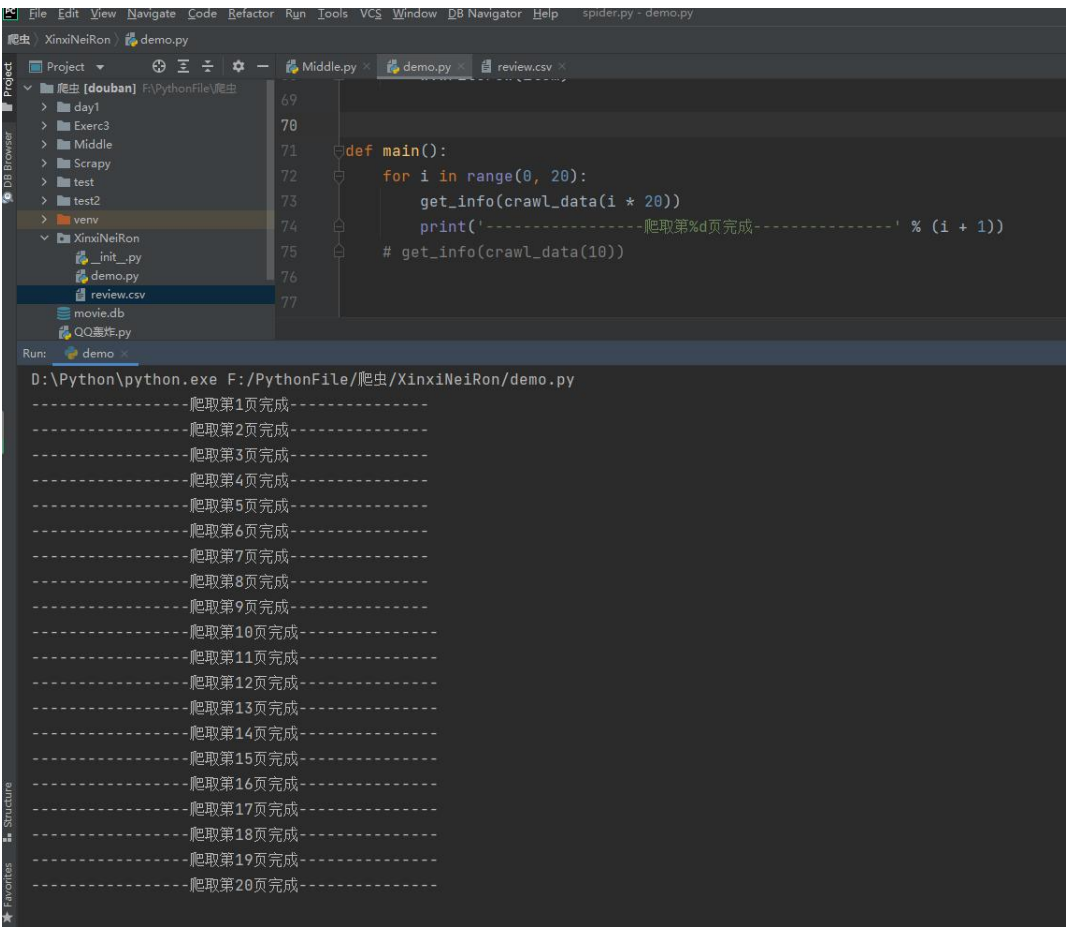
```

</div>
▼<div class="comment-item " data-cid="3024489759">
  ::before
  ▶<div class="avatar">...</div>
  ▼<div class="comment">
    ▼<h3>
      ▶<span class="comment-vote">...</span>
      ▼<span class="comment-info">
        <a href="https://www.douban.com/people/2099122/" class>Alan</a>
        <span>看过</span>
        <span class="allstar40 rating" title="推荐"></span> == $0
        <span class="comment time " title="2021-09-03 00:54:40"> 2021-09-
        </span>
      </span>
    </h3>
    ▼<p class="comment-content">

```

● 爬取数据

爬取 20 页评论数据，如下。



The screenshot shows an IDE with a project named '爬虫' (Crawler) containing a sub-project 'XinxiNeiRon'. The file 'demo.py' is open, showing a `main()` function that loops from 0 to 20, calling `get_info(crawl_data(i * 20))` and printing progress. The console output shows the script running successfully, printing 20 lines of '-----爬取第%d页完成-----' (Crawling page %d completed).

保存的 csv 文件如下所示，分别展示了每一条评论的内容和用户评价。



The screenshot shows a CSV file with two columns: 'review' (comment content) and 'rating' (user evaluation). The 'review' column contains detailed text about the movie 'The Great Wall' (《长城》), discussing its plot, cast, and production quality. The 'rating' column contains numerical ratings, mostly 10.

2、对评价数据进行分词（可以使用 Jieba、snownlp）

导入数据

```
In [9]: import pandas as pd

In [10]: df_data = pd.read_csv('review.csv', encoding = 'utf-8', names = ['content', 'score'])
df_data

Out[10]:
```

	content	score
0	【剧透！慎！】漫威系列中看的最莫名其妙的一部.....三观想不明白，父亲之前用十环犯下的恶报应在母...	较差
1	不不不还是不必了..... 杨紫琼对刘思慕说他长得跟陈法拉一模一样的时候我整个人在影院笑出声	还行
2	梁朝伟撑起全场，和男主对手戏时，演技大碾压！漫威也开始拍怪兽大战了.....	较差
3	其实我感觉，男主当爹，梁朝伟当儿子，效果会更好一些。	还行
4	在美国的电影院，看一部美国主流电影，听着大段大段的中文，旁边的美国人却要看着字幕，真的是很神奇...	推荐
...
415	男主好帅，梁朝伟太神了	推荐
416	上气原来姓徐，徐上气。刘思慕不愧是哈尔滨裔加拿大人，中英文都不带任何口音，尤其标准的哈尔滨话...	力荐
417	尚气的上映在asian American群体里可以说是个大事 甚至有人直接买光一个影院然后给...	力荐
418	比黑寡妇高出了一百个花木兰吧。	推荐
419	只敢刀刀向内的白左价值观，立基于此的剧情和人设一塌糊涂混乱不堪。当然动作戏和特效美工还是很赞...	还行

420 rows × 2 columns

使用结巴 jieba 分词

分词

2.1 jieba分词

```
In [11]: import jieba

In [12]: l_bow = []
for t in df_data.iloc[:,0]:
    segs = jieba.lcut(t, cut_all = False) # 对单句进行分段
    l_bow.append(segs) # 将分段后的句子添加到列表中

Building prefix dict from the default dictionary ...
Dumping model to file cache C:\Users\JUNDIN\1\AppData\Local\Temp\jieba.cache
Loading model cost 0.722 seconds.
Prefix dict has been built successfully.
```

```
In [13]: i = 0
for t in l_bow:
    print(i, end=' ')
    print(t)
    i += 1
```

分词结果

心，了，...，只，可惜，...，片子，从，选角，开始，就，争议，不断，...，一部，全片，有，一半，都，
在，说，普通话，的，好莱坞，大片，...，我们，竟然，是，那个，不能，在，大，银幕，上，看，的，...，
最后，日常，吼，一句，梁朝伟，真是，连，皱纹，都，好，有，魅力，...]
222 这不比，花木兰，好，上，个，卧虎藏龙，！，？？]
223 没有，任何，刻板，印象，...，全程，充满，了，对，中国，文化，的，尊重，和，理解，...，很难，想
象，能，有，一部，一半，以上，对白，都，是，标准，普通话，的，漫威，电影，...]
224 怀揣，0，期待，用，商业片，的，角度，去，看，倒，回馈，了，不少，惊喜，...，故事，
在，不，新，甚至，可以，说，老套，...，但是，作为，Marvel，的，电影，...，却，真正，
讲，中国，奇幻，故事，你，也，能，看到，导演，是，在，认真，做，功课，了，...，竹林，
和，...，弑父，...，是，李安式，的，...，而，兵器，与，情感，表达，...，又，是，那些，老港，片，
的，凝练，缩影，...，再，加上，山海经，的，...，意象，与，角色，...，真的，是，惊艳，...，一部，恰到好处，
的，全年龄，电影，...，中，西方，的，思维，的，故事性，达到，高度，统一，...，以及，梁朝伟，的，几
度，...，眼神，教科书，表演，...，彩蛋，再次，梦幻，联动，...，惊喜，不少，...，'，男主，一点，
都，不，开，啊，...]

使用 snownlp 分词

2.2使用snownlp 分词

```
In [14]: from snownlp import SnowNLP
l_bowl = []
for t in df_data.iloc[:,0]:
    s = SnowNLP(t)
    segs = s.words
    l_bowl.append(segs)
```

```
In [16]: i = 0
for t in l_bowl:
    print(i, end=' ')
    print(t)
    i+=1
```

```
6 ['推荐', '所有', '人', '去', '看', '尚气', '!', '作为', '爆米花', '电影', '!', '故事', '把', '西方', '个人', '价值', '和', '东方',
'文化', '气韵', '结合', '的', '非常', '顺畅', '!', '特效', '镜头', '制作', '精良', '!', '且', '中', '有', '我', '认为', '影史',
'最', '美', '的', 'XX!', '而且', '实话实说', '!', '这样', '的', '电影', '被', '好莱坞', '先', '拍', '出来', '!', '作为', '中国',
'电影', '人', '是', '有点', '惭愧', '的', '。']
7 ['...', '莫名其妙', '的', '内容', '乱七八糟', '的', '特效']
8 ['真的', '是', '超过', '预期', '了', '!', '没', '想到', '差不多', '特', '一半', '的', '对话', '都', '是', '中文', '!', '真是',
'没', '想到', '!', '还有', '故事', '的', '内核', '也', '很', '贴切', '我们', '的点', '!', '亲情', '兄弟', '姐妹', '情', '!', '真',
'的', '很', '吃', '我', '的点', '!', '喜欢', '!']
9 ['你', '将', '看到', '《', '花木', '兰', '》', '《', '卧', '虎藏', '龙', '》', '《', '别', '告诉', '她', '》', '《', '摘金', '奇', '缘', '》', '《',
'一代', '宗师', '》', '《', '功夫', '熊猫', '》', '以及', '《', '哥斯拉', '》', '等等', '组成', '的', '好莱坞', '有史以来',
'最', '眼花缭乱', '的', '一锅', '中西', '结合', '超级', '大', '乱', '炖', '!']
10 ['感觉', '在', '看', '花木', '兰', '!', '功夫', '熊猫', '!', '元素', '杂', '糅', '!', '自', '以为', '是', '的', '中国', '风',
'!', '花里', '胡哨', '的', '颜色', '!', '老', '套', '的', '故事']
11 ['梁', '朝', '伟', '倾', '情', '演绎', '思念', '亡', '妻', '的', '深情', '丈夫', '和', '误', '信', '诈骗', '电话', '的', '独',
'居', '老人']
12 ['梁', '朝伟', '在', '里面', '哪', '是', '什么', '反派', '!', '!', '不过', '是', '高高兴', '出门', '买', '菜', '回家', '却', '发现',
'爱人', '没', '了', '的', '绝望', '丈夫']
13 ['我', '一点', '没', '看', '出来', '上气', '有', '啥', '黑暗', '的', '过去', '啊', '!', '一天', '跟', 'Katy', '俩', '没', '心',
'没', '肺', '的', '样子', '!', '这', '人物', '曲线', '一点', '也', '不', '饱满', '!', '也', '不', '全', '是', '剪辑', '的', '结果',
'!', '演员', '没', '演出', '来']
```

3、在分词的基础上，利用词库对评价数据进行情感评分

Bosonnlp 先对每行中的分词进行情感计算并求总和得出一行句子的情感数值

基于bosonnlp词典的情感计算

```
In [17]: df_score = pd.read_csv('BosonNLP_sentiment_score.txt', sep=' ', names=['word', 'score'], header=None )
print(df_score)
```

```
      word  score
0      最尼玛 -6.704000
1      扰民 -6.497564
2      fuck... -6.329634
3      RNM -6.218613
4      wcnmlgb -5.967100
...
114761 prada667 6.375039
114762 如虎添翼 6.375039
114763 订购 6.375039
114764 富婆团 6.375039
114765 赖世荣 6.375039
```

[114766 rows x 2 columns]

```
In [20]: l_seg1 = []
ss = 0
for b in l_bow: # b是一个评论句子
    score = 0
    for w in b: # w 是其中分词的一个词
        for s in df_score.iteruples():
            if w == s[1]: # 找到相同的字段
                score = score + s[2] # 对应的情感数值
                break
    # print(score, end=' ')
    l_seg1.append(score)
```

基于 **Bosonnlp** 词典的情感计算结果

SnowNLP 情感分析也是基于情感词典实现的，其简单的将文本分为两类，积极和消极，返回值为情绪的概率，越接近 1 为积极，接近 0 为消极。

想「不」剧「透」！「父」的「漫」威「系」列「中」「香」的「最」在「真」名「妙」的「一」部「……」三「」观「环」当「香」儿「子」的「面」前「上」打「死」母「亲」的「门」派「的」手「下」母「亲」的「身」上「父「亲」用「十」血「债」血「偿」的「来」受「了」反「派」的「怪」物「的」影「响」强「逼」母「亲」的「故」乡「莫「救」在「母「亲」当「年」种「下」的「恶」果「要「血」债「血」偿「杀」了「父「亲」……」母「亲」的「一」切「的」恨「更「说「不」过「的」其「实」是「刚」开「始」父「亲」手「下」去「抢」母「亲」留「给」兄「妹」的「遗」物「那「是「打「开「走「入「村「庄「大「门」的「线「索」找「孩「子「回「来「就「找「孩「子「将「那「那「那「追「杀「没「区「别「兄「妹「真「名「妙「的「看「完「反「正「最「后「给「父「亲」也「报「仇「了「将「十「环「继承「和「给「了「儿「子「了「不「梁「朝「伟「联「合「杀「了「父「亲」反「派」的「给「父「亲」报「仇「了「了「不「过「香「出「来「了「花「了「不「梁「朝「伟「打「造「出「了「布「景「很「美「很「好「看「特「效「做「的「很「大「打「戏「依「旧「很「多「大「量「中「文「台「词「瞬「间「以「为「自「己「在「国「内「电「影「院」」

1「不」不「不」还「是「不「必「了「……」杨「紫「琼」对「刘「思「慕」说「他「长「得「跟「陈「法「拉」一「模「一「样」的「时「候「我「整「个「人「左「影「象」笑「出「声」

2「梁「朝「伟」播「起「全「场」和「男「主「对「手「戏「时「演「技「大「大「碾「压」了「漫「威」也「开「始「拍「怪「兽「大「战」了「……」

3「其「实「我「感「觉「到「男「主「当「爹「的「梁「朝「伟」当「儿「子」的「效果「会「更「好「一些」

「在「美「国」的「电「影「院」中「看「到「美「国「主「演「电「影「的「香「港「大「佬「本「会」的「中「文」

基于 Snownlp 的情感分析结果

1.0 0.8434243065022546 0.896175173367221 0.9382135437457467 0.9999973434435147 0.5411525247081855 0.9999999977187899 0.042
809361359257836 0.999989443781517 0.9999992165602503 0.9941665445986734 0.9998434035942745 0.8716017951587698 0.999890369247
4193 0.9999736595850125 0.33740335621729445 0.9394363487546916 0.972530599656153 0.9866556848609973 0.9998691300404706 1.0
0.8434243065022546 0.896175173367221 0.9382135437457467 0.9999973434435147 0.5411525247081855 0.9999999977187899 0.042809361
359257836 0.999989443781517 0.9999992165602503 0.9941665445986734 0.9998434035942745 0.8716017951587698 0.9998903692474193
0.9999736595850125 0.33740335621729445 0.9394363487546916 0.972530599656153 0.9866556848609973 0.9998691300404706 0.99748456
98254331 0.9747374931433581 0.999999906431168 1.0 0.9999513735762411 0.869340955710838 0.9958650388442201 0.9999360086638
378 0.9971744183686536 0.9768429150801266 0.9155540569078949 0.021199808079638505 0.999999999879317 0.9628469663779651 0.9
99999583381336 0.9961467941743506 0.8450818405286782 0.9994251276140314 1.0 0.999991421655637 0.21301367209596067 0.5922
954614034758 0.9504642888549727 0.827129641865894 0.7699447249904937 0.999999994669635 0.999999800406638 1.0 0.355304382
72723075 0.9999999999919773 0.9466099672508748 0.8066401737069598 0.999999999997542 0.7830543198525399 0.9053846015094014
0.9702838867822937 0.7478703231998379 0.9961104357818747 0.9721485599011247 0.9999902385522209 0.9995417619660798 0.30768615
146877965 0.999999988978768 0.8560760992115735 0.08978918642357891 0.9999960731351347 0.9031346028837043 0.9992932578985539
0.9161222313784824 0.9999478581842475 0.999999999942968 0.999975189758554 0.999999996456022 0.9999919357397129 0.985302669
2842559 0.018431214608679025 0.8192894123319076 0.9975735935539146 0.8115502002192323 0.999999758370234 0.9993604269464793
0.368249363832295 0.9999997799331486 0.999999999999143 0.9867433517165547 0.999999997527351 0.9643888149567564 0.9941294336
299432 0.056656889795661414 0.9972012154471944 0.9999999996577627 0.9990248432137067 1.0 0.8708197180432481 0.963165699431
7493 0.9999948671883053 0.9959501914286369 0.4198217536626575 0.1329364186028662 1.0 0.9959501914286369 0.4198217536626575
1.0 0.999994364246537 0.999866890858942 1.0 0.999999758370234 0.9999999655041568 0.9999994501066578 0.9999997799331486
0.996525119068169 0.9999999999999996 1.0 0.9507145512310418 0.9412680392796318 0.1329364186028662 0.999976932043384 0.997
0990899768117 0.99988845203923 1.0 1.0 0.9999942325854104 0.9936621485235316 1.0 1.0 0.8381714884963163 0.999999999783
5811 1.0 0.9725152947828228 0.38229653402590635 0.05729368649477895 0.9930016748217906 0.999999999999452 0.31982044489254
07 0.9922830214426002 0.8460186392360579 0.7160292799456651 1.0 0.9999999999995166 0.0007128324866848557 0.991288966657895
7 0.9999689385242138 0.43184245178034086 0.9997044991151756 0.9999996771204135 0.5438259652718098 0.5790720615437331 0.634
2546484425678 0.9837008370656131 0.9999993359315646 0.9960293497733906 0.999999999979687 0.999999985815875 0.9987652750378
244 0.9131507565648229 0.9986234787391207 0.9031307253126337 0.9968894148453239 0.8293561721738222 0.9999999999962113 0.99
35203344494872 0.9999999999999836 0.9999237229920026 0.9649446188131584 1.0 0.9999999988142434 0.0001651271464917503 0.999
9999999999998 0.17160759266655612 0.9850837094718516 0.9999602101042109 0.9999980083087401 0.9999237942722558 0.999999607347
0095 0.9911505511243713 1.0 0.9999999999942186 0.9999996299534027 0.7194444444444446 0.13314680229950815 0.996105146676380
6 0.596486202959828 0.9999999922824027 0.9651072890883424 0.9999860868901703 0.6234577089090972 0.9791143730971849 0.84371
52884588428 0.9863244496837364 0.9998492213505544 0.999999999978338 0.9999299325669253 0.7929059586125223 0.9999977555008452
0.7222499739114551 0.5363537462550282 0.7774061061617753 0.9999975754662678 0.9994517031709452 0.8844759160061365 0.99861586
55822861 0.999999999999390399 0.9797344432448709 0.9995792891562693 1.0 0.9999994484151473 0.9999999998644479 0.927833822413
0797 0.42086589820895615 0.999999999999973 0.18490650759925276 0.10978739439094831 0.9998733215684041 0.9997461255663346
0.999977844221345 0.9999540635586663 0.1450771767647212 0.9974209624793147 1.0 0.9998946194997964 0.999999999973095 1.0
0.9984920199045242 0.9994505614361607 0.9773831855976763 0.9555804587602811 0.9999482671451999 0.5262327818078083 0.58666246
6766194 0.9999999603691274 0.3011416646919197 0.8065487303571107 0.4479678413036058 0.9269038210033553 0.999999999972753
0.99999132646489 0.9999999999999999 0.11893351500750501 0.1314814046815388 0.9977609941316639 0.9999976696651074 0.7189189

基于 cnsenti 中文情绪情感分析库

- words 文本中词语数
- sentences 文本中句子数
- pos 文本中正面词总个数
- neg 文本中负面词总个数

```
In [15]: l_seg1 = []  
l_seg2 = []  
senti = Sentiment()  
for t in df_data.iloc[:,0]:  
    seg1 = senti.sentiment_count(t)  
    seg2 = senti.sentiment_calculate(t)  
    l_seg1.append(seg1)  
    l_seg2.append(seg2)
```

```
In [16]: l_seg2
```

```
Out[16]: [('sentences': 5, 'words': 229, 'pos': 140.0, 'neg': 984.0),  
( 'sentences': 1, 'words': 29, 'pos': -21.0, 'neg': 0.0),  
( 'sentences': 2, 'words': 22, 'pos': 0, 'neg': 0),  
( 'sentences': 1, 'words': 16, 'pos': 0, 'neg': 0),  
( 'sentences': 1, 'words': 34, 'pos': 0, 'neg': 0),  
( 'sentences': 1, 'words': 19, 'pos': 0, 'neg': 0),  
( 'sentences': 4, 'words': 58, 'pos': 5.0, 'neg': 2.0),  
( 'sentences': 1, 'words': 7, 'pos': 0, 'neg': 9),  
( 'sentences': 4, 'words': 43, 'pos': 1.0, 'neg': 18.0),  
( 'sentences': 2, 'words': 47, 'pos': 0.0, 'neg': 34.0),  
( 'sentences': 1, 'words': 23, 'pos': 0, 'neg': 12),  
( 'sentences': 1, 'words': 15, 'pos': 9, 'neg': 0),  
( 'sentences': 1, 'words': 21, 'pos': 44.0, 'neg': 0.0),  
( 'sentences': 1, 'words': 41, 'pos': 0.0, 'neg': -16.0),  
( 'sentences': 1, 'words': 50, 'pos': 63.0, 'neg': 0.0),  
( 'sentences': 1, 'words': 7, 'pos': 0, 'neg': 0),  
( 'sentences': 1, 'words': 28, 'pos': 0, 'neg': 0),  
( 'sentences': 1, 'words': 14, 'pos': 0, 'neg': 0),  
( 'sentences': 1, 'words': 61, 'pos': 0.0, 'neg': 245.0),  
( 'sentences': 2, 'words': 45, 'pos': 10, 'neg': 0),
```

Term [1]


```

({'sentences': 3, 'words': 55, 'pos': 0.0, 'neg': 4.0},
 {'sentences': 1, 'words': 22, 'pos': 7, 'neg': 0},
 {'sentences': 1, 'words': 23, 'pos': 15.0, 'neg': 15.0},
 {'sentences': 1, 'words': 28, 'pos': 53, 'neg': 8},
 {'sentences': 2, 'words': 27, 'pos': 0, 'neg': 0},
 {'sentences': 5, 'words': 119, 'pos': 180.0, 'neg': 72.0},
 {'sentences': 1, 'words': 23, 'pos': 15.0, 'neg': 15.0},
 {'sentences': 1, 'words': 28, 'pos': 53, 'neg': 8},
 {'sentences': 5, 'words': 119, 'pos': 180.0, 'neg': 72.0},
 {'sentences': 3, 'words': 77, 'pos': 0.0, 'neg': 52.0},
 {'sentences': 2, 'words': 40, 'pos': 21, 'neg': 0},
 {'sentences': 5, 'words': 180, 'pos': 300.5, 'neg': 378.0},
 {'sentences': 1, 'words': 65, 'pos': 281.0, 'neg': 61.0},
 {'sentences': 4, 'words': 118, 'pos': 57.0, 'neg': 185.0},
 {'sentences': 4, 'words': 95, 'pos': 125.0, 'neg': 43.0},
 {'sentences': 5, 'words': 84, 'pos': 165.0, 'neg': 27.0},
 {'sentences': 5, 'words': 81, 'pos': 10.0, 'neg': 27.0},
 {'sentences': 7, 'words': 109, 'pos': 71.0, 'neg': 103.0},
 {'sentences': 6, 'words': 182, 'pos': 54.0, 'neg': 438.0},
 {'sentences': 2, 'words': 22, 'pos': 1, 'neg': 7}).

```

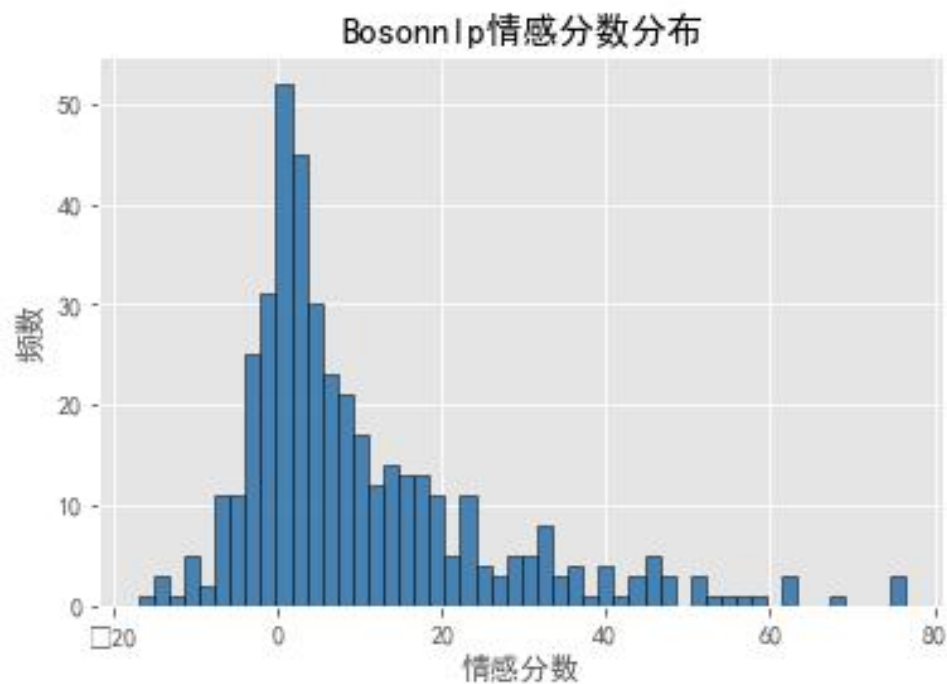
```

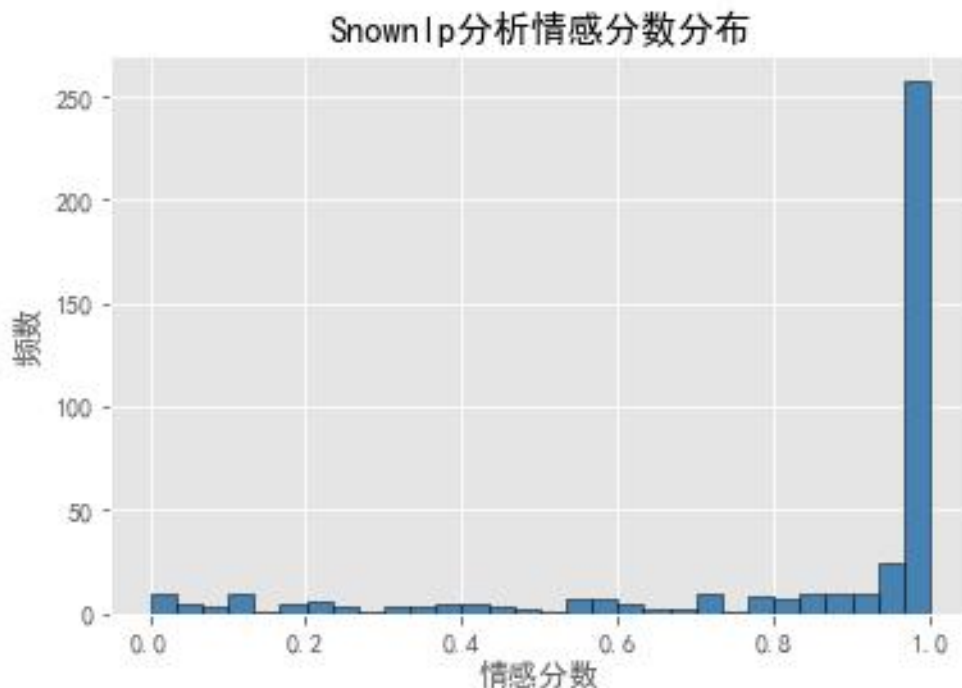
In [17]: l_seg1
Out[17]: [{'words': 229, 'sentences': 5, 'pos': 6, 'neg': 10},
 {'words': 29, 'sentences': 1, 'pos': 1, 'neg': 0},
 {'words': 22, 'sentences': 2, 'pos': 0, 'neg': 0},
 {'words': 16, 'sentences': 1, 'pos': 0, 'neg': 0},
 {'words': 34, 'sentences': 1, 'pos': 0, 'neg': 0},
 {'words': 19, 'sentences': 1, 'pos': 0, 'neg': 0},
 {'words': 58, 'sentences': 4, 'pos': 1, 'neg': 1},
 {'words': 7, 'sentences': 1, 'pos': 0, 'neg': 2},
 {'words': 43, 'sentences': 4, 'pos': 1, 'neg': 3},
 {'words': 47, 'sentences': 2, 'pos': 0, 'neg': 2},
 {'words': 23, 'sentences': 1, 'pos': 0, 'neg': 1},
 {'words': 15, 'sentences': 1, 'pos': 1, 'neg': 0},
 {'words': 21, 'sentences': 1, 'pos': 2, 'neg': 1},
 {'words': 41, 'sentences': 1, 'pos': 0, 'neg': 1},
 {'words': 50, 'sentences': 1, 'pos': 2, 'neg': 0},
 {'words': 7, 'sentences': 1, 'pos': 0, 'neg': 0},
 {'words': 28, 'sentences': 1, 'pos': 0, 'neg': 0},
 {'words': 14, 'sentences': 1, 'pos': 0, 'neg': 0},
 {'words': 61, 'sentences': 1, 'pos': 1, 'neg': 3},
 {'words': 45, 'sentences': 3, 'pos': 1, 'neg': 0}].

```

4、对比不同工具的分词结果和情感分析结果，进行分析和总结。（**snownlp**, **cnssenti**, **bosonnlp**）

基于 **Bosonnlp** 词典和 **Snownlp** 的情感计算结果绘成直方图





BosonNLP 情感分析是基于词典对列表数据进行逐个匹配，并记录匹配到的情感词分值。最后，统计计算分值总和，如果分值大于 0，表示情感倾向为积极的，如果小于 0，则表示情感倾向为消极的。从 BosonNLP 分析结果直方图中可看出大部分数据是大于 0 的，说明大都数评论呈好感、正向积极的；从 Snownlp 情感分数直方图可以看出，大部分数据是接近于 1，也表示说明大部分评论是正向积极的。从 cnsenti 情感分析库的输出来看，“pos”的值要大于“neg”的情况居多，说明正向词居多，也表明评论的大多数态度是积极的或者中性的。

BosonNLP 首先对文本进行分句、分词，将分词好的列表数据对应 BosonNLP 词典进行逐个匹配，记录匹配到的情感词分值。统计每一句的分值总和来判断这句话的态度；snownlp 和 csnenti 可以针对单个句子进行情感分析数值计算；BosonNLP 和 Snownlp 都通过数值来反映情感状况，而 cnsenti 则显示的分析文本的词性正负情况，并且支持多种情绪统计分析。

【思考题】

1. 查阅文献，思考还有哪些情感分析方法，以及情感分析面临的问题有哪些？。

文本情感分析主要有三大任务，即文本情感特征提取、文本情感特征分类以及问文本情感特征检索与归纳。主要分为两类：

- 基于情感词典的方法

人工构建情感词典

自动构建情感词典

- 基于机器学习的方法

朴素贝叶斯

最大熵

SVM 分类器

准确的情绪分析的主要障碍一直是自然语言，现在仍然是，原因有很多。尽管在情感分析上，越来越多的研究取得了进展，但处理文本中的“‘affective phenomena’”，如主体性、aspects、情感、情绪、语气、态度和感受，已被证明是一个复杂的、跨学科的问题，远远没有得到解决。必须考虑许多参数，如作者的个人资料、文本类型、样式、域、文档来源、目标语言和最终应用的目标。公开的实验结果(通常在相对有利的环境中获得)与系统在真实环境中获得的结果之间也存在差距。