结课报告------基于TextRank与情感分析的电影多维度评判

小组：挖掘多人

成员：

- 刘金田（3220190844）
- 吴嘉豪（3220190894）
- 易鑫（3120191066）
- 李祥潮（3220190830）
- 孙玥（3120191046）

1 课程项目内容与意义

1.1 内容

> 本项目旨在对微博、豆瓣、新闻等相关文本资源进行收集、分类、统计、分析与可视化，并针对去年过年时期的《流浪地球》进行相关评论的情感分析，这部电影在去年的大众网评曾经出现过较大的波动与两极分化，故此希望通过相关数据挖掘技术与情感分析算法找出大众的对该片的情感意见。
> 本项目通过爬虫等方法对网络上该电影的相关文本评论加以收集，采用词频统计、TextRank算法实现关键词等词向量的表示，通过TextRank算法对爬取的新闻进行重要性排序从而得到电影从上映前、放映时到上映后的事件脉络，使用LSTM等深度学习算法完成对文本评论的情感判断，并使用直方图、词云等可视化方法展示该项目得到的相关结论，并针对高频词汇，使用Apriori算法或者FD-tree对其进行频繁模式挖掘和关联规则分析。

1.2 意义

> 微博等网络评论往往能在一定程度上代表了社会公众对于某些突发事件、热点事物的看法与评价，大多数的网络使用者都会倾向于使用微博等社交网络工具表达对于事件、人物、电影等的看法，同时人们也会在浏览微博的同时被微博的相关舆论趋势所引导。鉴于此，微博评论等往往能够对有关公司、相关人员产生极大的影响与作用。如何利用好这些评价结果，分析得到网络上的主流意见，并加以可视化、分析乃至利用是一个更加值得关注的问题。

2 数据获取及预处理

2.1 数据来源

> 使用爬虫的数据挖掘手段对有关《流浪地球》的相关文本评论进行收集,包括豆瓣电影短评、新闻与微博相关电影评论。
> **电影评论数据收集**:
> 使用了selenium库以尽量避免遭到限制，利用当前目录下有谷歌浏览器的webdriver进行登录与爬取，爬取的影评数据包括: MTime时光网影评短评 (包括最新与最热)共计约540条，豆瓣短评 (包括最新与最热)共计600条，豆瓣长影评爬取了标题与简单文章内容截取共计18940条。其中鉴于豆瓣短评受限仅能看到部分页评论因此额外爬取了时光网短评以作弥补。同时在新浪微博，爬取关于流浪地球电影热门微博下的评论，和流浪地球上映三个月内流浪地球话题下的原创微博。

> **电影新闻数据收集**:
> 使用了相似的方法爬取了有关于流浪地球的新闻报道约三百余条，每条按照固定格式进行命名并且对具体新闻的报道进行了保存。

2.2 数据说明

- MTime.csv 时光网影评短评（包括最新和最热），共540条记录

  > 数据各字段描述- text:短评文本内容

- douban_duanping.csv 豆瓣短评（包括最新和最热），共600条记录

  > 数据各字段描述- text:短评文本内容

- douban_changping.csv 豆瓣影评， 共18940条记录

  > 数据各字段描述- title:影评标题；text:影评文本内容

- xinlang.csv 新浪娱乐相关新闻， 共308条记录

  > 数据各字段描述- title:短评标题；

- weibo_comment.csv:关于流浪地球电影热门微博下评论 共5000条

  > 数据各字段描述- time:发布时间；text:评论文本内容；uid:用户id;like_count:获得点赞数；username:用户名；following:关注数；followed:粉丝数；gender:性别

- weibo_topic.csv:流浪地球上映三个月内流浪地球话题下的原创微博，共3000条

  > 数据各字段描述- 微博id;发布者姓名；发布者性别；发布者地区；发布者关注数；发布者粉丝数；微博正文；原始图片url;发布位置；发布时间；发布工具；点赞数；转发数；评论数；

2.3 数据预处理

对豆瓣评论数据，合并长短评并删除空评论、
对新浪娱乐相关新闻数据，删除仅含有标题的新闻和过短的新闻。
对新浪微博评论数量合并并删除空评论和部分无意义评论。

对空值和无意义数据进行数据预处理之后，将所有评论/新闻放在一个txt中，一行是一条评论/新闻。

3 模型以及相关算法介绍

3.1 模型

- word2vec模型
  word2vec是google开源的一款用于词向量计算的工具，也是一种语言算法模型。通过使用word2vec模型，可用高维向量表示词语，建立词向量和索引表。
- LSTM模型
  LSTM(Long-Short Term Memory,长短期记忆人工神经网络)是一种特定的循环神经网络，具有较好的处理长序列数据的能力。将一个句子中的词向量输入到LSTM神经网络中，可用于判断句子的情感极性，分析大众的评论是好评或是差评。
  LSTM输出长度为50向量，3个单元全连接层+softmax层输出结果.

3.2 其他重要算法

- TextRank算法
  TextRank算法是一种抽取式的无监督的文本摘要方法。用于为文本生成关键字和摘要，本实验通过使用TextRank算法完成词频分析与关键词统计，并基于TextRank算法对爬取的新闻进行重要性排序，得到电影从放映前到放映后的事件脉络。

- Apriori算法
  Apriori算法是种挖掘关联规则的频繁项集算法，通过使用Apriori,获取频繁模式及关联规则。

4 课程项目实验过程

4.1 提取关键词

使用TextRank算法提取关键词

- 对每个句子进行分词和词性标注处理,并根据*stopwords.txt*过滤停用词，只保留了部分词性的词（如名词、动词、形容词等）
- 构建候选关键词图G=(V,E),其中V为节点集，由上一步保留后的候选关键词组成，然后采用共现关系构造任两节点之间的边，当且仅当两个节点对应的词汇在长度为K的窗口中共现，两个节点之间存在边，K表示窗口大小，即最多共现K个单词。
- 根据TextRank公式，迭代传播各节点的权重，直至收敛

$$WS(V_i) = (1-d) + d \times \sum_{v_j \in \ln(V_i)} \frac{w_{ji}}{\sum_{V_k \in O(v_j)} w_{jk}} WS(V_j)$$

- 对节点权重进行倒序排序，得到最重要的x个单词，作为候选关键词
- 由上一步得到最重要的x个单词，在原始文本中进行标记，若形成相邻词组，则组成多词关键词。
- 最后分别将各平台评论关键词保存于txt文件，一行保存一条记录的关键词。

4.2 由新闻分析电影上映前后事件脉络

使用TextRank对新闻生成摘要

- 对每条新闻内容分进行分句。
- 将新闻内的每个句子分别看做是一个节点，如果两个句子有相似性，那么认为这两个句子节点之间存在一条无向有权边，权重为相似度大小。
- 其中相似度计算：
  -- 对任意两句话a,b；
  -- 对a,b进行分词，得到两个分词结果列表words_a,words_b。合并两列表为words；
  -- 分别统计words中每个单词在a,b两句话中出现的次数，记为n,m;
  -- 将n,m对应元素相乘后相加，得到ans;
  -- 则相似度为

$$Similarity = \begin{cases} 0, & ans = 0 \\ \dfrac{ans}{ln(len(words\_a)) + ln(len(words\_b))}, & others \end{cases}$$

- 最后根据分数的高低，将所有句子按照分数从高到低排列，取前k个句子为最终被选为摘要的句子。
  使用TextRank对新闻重要性分析：
- 将每条新闻看做一个节点，边的权重为两新闻之间的相似性。
- 其中，相似性计算：
  -- 对任意两新闻a,b，分词后统计词频，并为每篇新闻维护一个词频字典 dic_a,dic_b(键为单词，值为词频)
  -- 求新闻ab词汇的交集words,遍历words,words中每个词有一个得分score,求所有的score,得到两

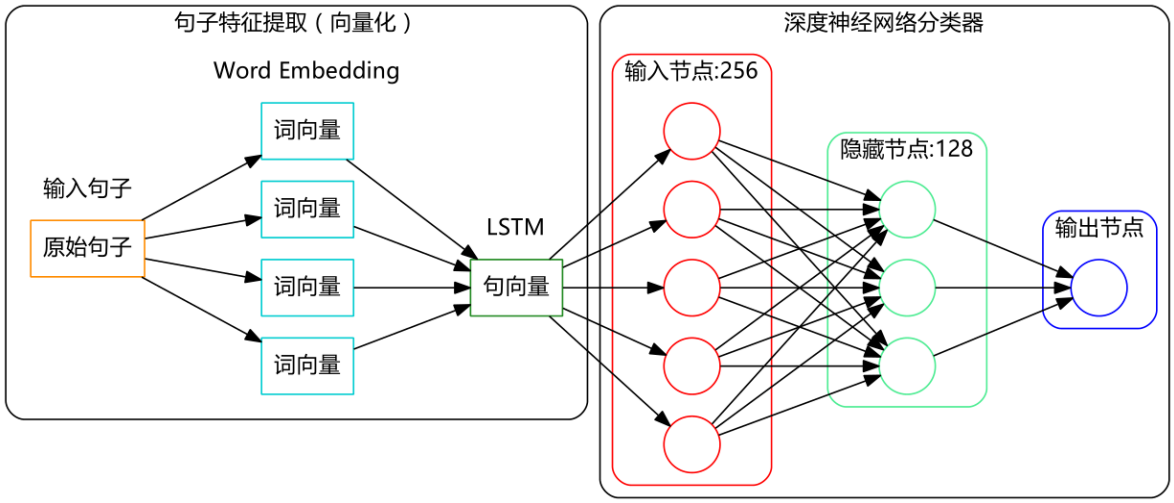
新闻的相似度。对words中每个单词word，

$$score = round(\tanh \frac{dic\_a[word]}{dic\_b[word]})$$

- 最后根据分数高低，得到新闻的重要性排序。

4.3基于LSTM的情感分析

本实验情感分析部分使用word2vec将词语转化为高维向量后，输入到LSTM中，其结构图如下所示：



- 首先训练word2vec模型，将每个词映射为1个100维向量，训练的主要参数有：

```
# set parameters:
cpu_count = multiprocessing.cpu_count()  # 控制训练的并行数
vocab_dim = 100  # 特征向量的维度
n_iterations = 4  # 迭代的次数
n_exposures = 10  # 可以对字典做截断. 词频少于min_count次数的单词会被丢弃掉
window_size = 7  # 窗口大小，表示当前词与预测词在一个句子中的最大距离是多少
```

- 输入文本后，将文本划分为很多句子，对每个句子使用结巴分词进行分词，根据*stopwords.txt*滤掉了停用词。
- 将句子分好词并转换为高维向量，得到词向量的集合，将集合展平，得到一个一维长向量，为LSTM的输入值，LSTM输出长度为50向量，3个单元全连接层+softmax层输出结果。下图为训练截图和测试截图

```
16448/16870 [===========================>.] - ETA: 0s - loss: 0.6025 - acc: 0.9487
16480/16870 [===========================>.] - ETA: 0s - loss: 0.6026 - acc: 0.9487
16512/16870 [===========================>.] - ETA: 0s - loss: 0.6028 - acc: 0.9485
16544/16870 [===========================>.] - ETA: 0s - loss: 0.6028 - acc: 0.9484
16576/16870 [===========================>.] - ETA: 0s - loss: 0.6028 - acc: 0.9485
16608/16870 [===========================>.] - ETA: 0s - loss: 0.6027 - acc: 0.9486
16640/16870 [===========================>.] - ETA: 0s - loss: 0.6027 - acc: 0.9486
16672/16870 [===========================>.] - ETA: 0s - loss: 0.6027 - acc: 0.9485
16704/16870 [===========================>.] - ETA: 0s - loss: 0.6027 - acc: 0.9485
16736/16870 [===========================>.] - ETA: 0s - loss: 0.6026 - acc: 0.9486
16768/16870 [===========================>.] - ETA: 0s - loss: 0.6026 - acc: 0.9487
16800/16870 [===========================>.] - ETA: 0s - loss: 0.6025 - acc: 0.9487
16832/16870 [===========================>.] - ETA: 0s - loss: 0.6025 - acc: 0.9487
16864/16870 [===========================>.] - ETA: 0s - loss: 0.6025 - acc: 0.9487
16870/16870 [==============================] - 28s 2ms/step - loss: 0.6026 - acc: 0.9487
Evaluate...
3104/4218 [=====================>........] - ETA: 0s
3232/4218 [=====================>........] - ETA: 0s
3360/4218 [======================>.......] - ETA: 0s
3488/4218 [======================>.......] - ETA: 0s
3616/4218 [=======================>......] - ETA: 0s
3744/4218 [=======================>......] - ETA: 0s
3872/4218 [========================>.....] - ETA: 0s
4000/4218 [========================>.....] - ETA: 0s
4128/4218 [=========================>....] - ETA: 0s
4218/4218 [==============================] - 2s 457us/step
Test score: [0.6474404856739252, 0.9030346135891913]

进程已结束，退出代码 0
```

4.4 基于哈工大ltp的语义情感分析

由于情感分析中LSTM模型对短评效果不好，微博评论一般都较短，因此，使用基于哈工大ltp的语义情感分析对微博短评论进行情感分析。

- 输入文本，将文本以标点符号为界限划分为很多句子，对每个句子分词，计算每个句子包含的情感词个数和情感词列表，然后过滤掉不包含情感词的句子。
- 对每个句子：

  分词后得到每个词的词性，根据分词列表和词性列表进行该句子的依存句法分析，它返回一个列表，列表中每一个元素代表：单词索引，单词，单词词性，单词依存的单词和词性，索引和依存关系然后基于上述得到的列表和分词列表词性列表计算句子中每一个词（不同依存关系-依存儿子和依存父亲）的字典映射
- 最终流程：

  对过滤后的每个句子分词列表遍历，若单词是情感词，则根据情感词典得到其对应分数，接着若该词存在依存父亲和儿子，该依存父亲或儿子是修饰词且词性不以w或u开头，则在修饰词典找到其权重，与上述分数相乘，迭代后得到该词的最终分数，每个句子所有词分数相加得到该句子的最终分数。
- 文章最后分数等于所有句子分数之和/句子总数

4.5 频繁项集挖掘

- 读取4.1 提取的关键词数据集。

- 使用Apriori算法进行频繁项集挖掘
  -- 初始通过单遍扫描数据集，确定每个项的支持度，得到所有频繁1-项集的集合F1.
  -- 迭代地使用上一步得到的k-1项集，产生新的候选k项集。
  -- 扫描数据计算候选频繁k项集的支持度。
  -- 计算候选项的支持度，删除支持度低于阈值的数据，得到频繁k项集。
  -- 当没有新的频繁项集产生，算法结束。

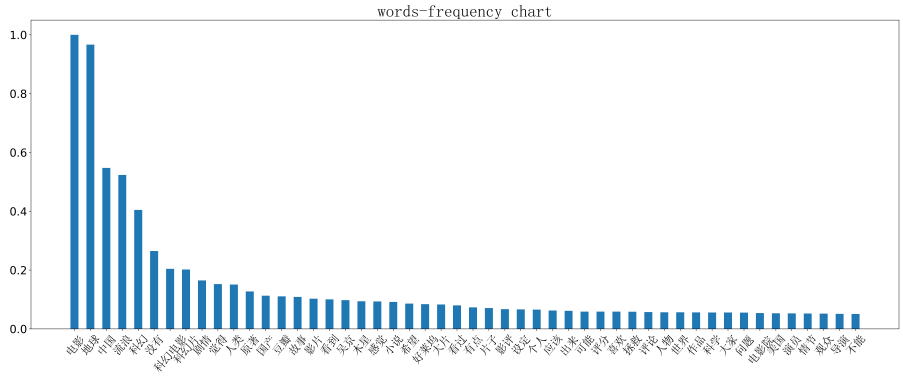
- 根据所有的频繁项集进行关联规则挖掘

```
def Generate_Rule(self, L, support_data, min_confidence):
    '''
    参数：所有的频繁项目集，项目集-支持度dic，最小置信度
    '''
    rule_list = []
    sub_set_list = []
    for i in range(len(L)):
        for frequent_set in L[i]:
            for sub_set in sub_set_list:
                if sub_set.issubset(frequent_set):
                    conf = support_data[frequent_set] / support_data[sub_set]
                    # 将rule声明为tuple
                    rule = (sub_set, frequent_set-sub_set, conf)
                    if conf >= min_confidence and rule not in rule_list:
                        rule_list.append(rule)
            sub_set_list.append(frequent_set)
    return rule_list
```

5 实验的结果及可视化

5.1 关键词

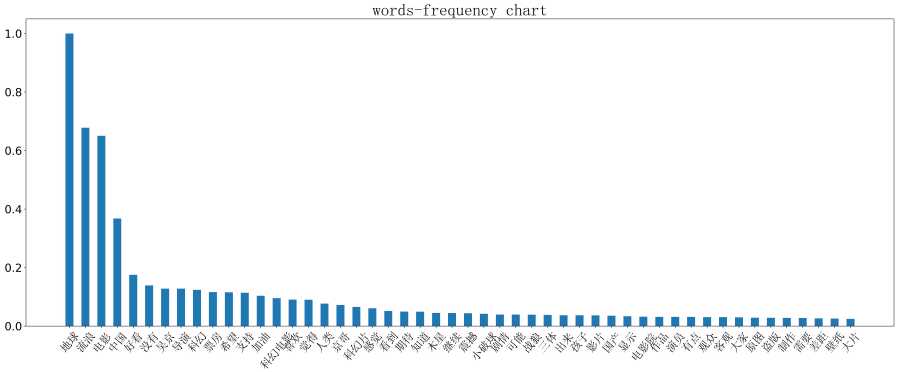
- 豆瓣平台




导演
人类
可能
电影院
有点
国产
流浪
剧情
评论
没有
觉得
看过
不能
美国
影片
希望
影评
演员
应该
世界
好莱坞
科幻
电影
地球
小说
评分
人物
情节
作品
看到
豆瓣
大家
大片
拯救
感觉
中国
科学
问题
科幻电影
片子
设定
原著
木星
人
喜欢
出来
吴京
科幻片
观众
故事

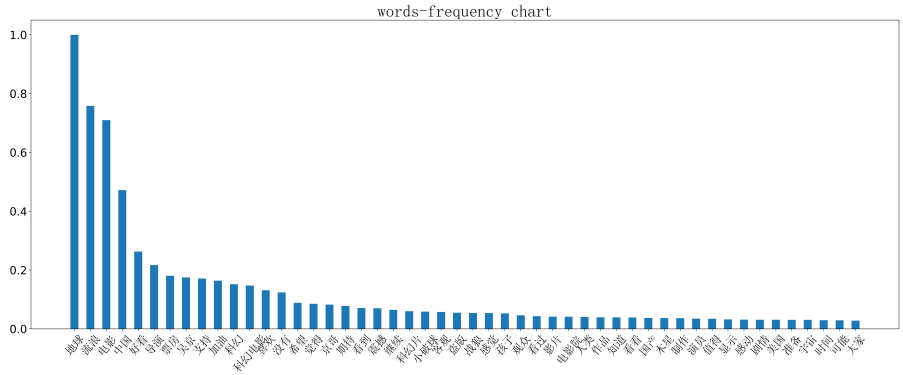
- 微博平台

1. 所有用户：


words-frequency chart
1.0
0.8
0.6
0.4
0.2
0.0

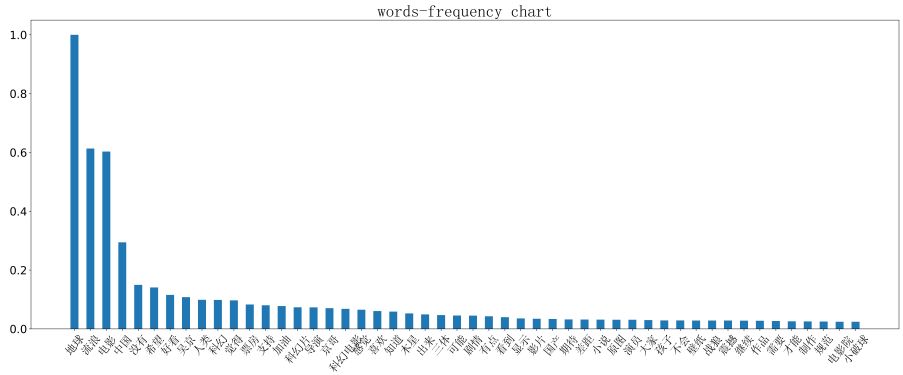

科幻电影
小破球
大家
中国
好看
制作
剧情
加油
孩子
感觉
盗版
木星
期待
没有
作品
地球
科幻
原图
觉得
科幻片
战狼
需要
有点
看到
吴京
观众
电影
客观
显示
壁纸
流浪
影片
震撼
电影院
票房
喜欢
差距
可能
知道
导演
人类
演员
国产
支持
希望
京哥

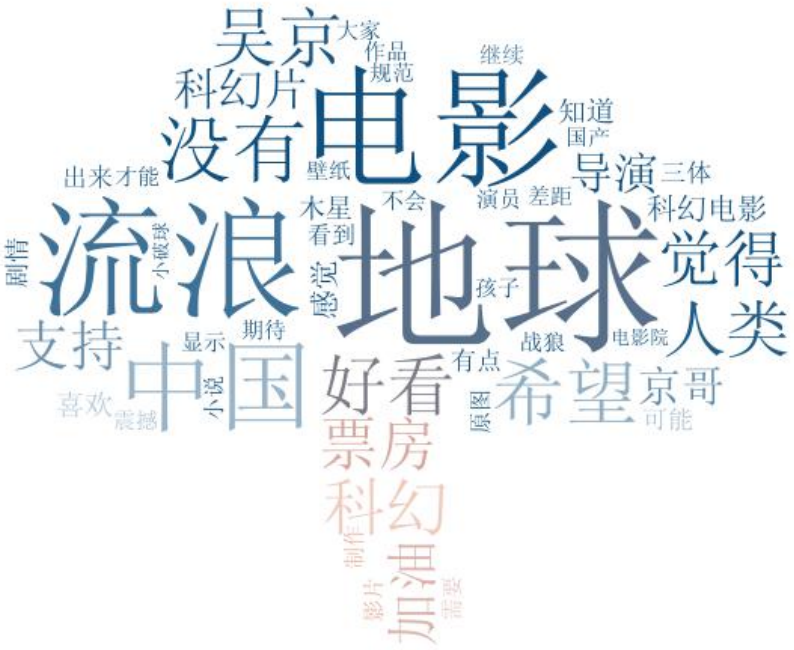
2. 男性用户


words-frequency chart
1.0
0.8
0.6
0.4
0.2
0.0

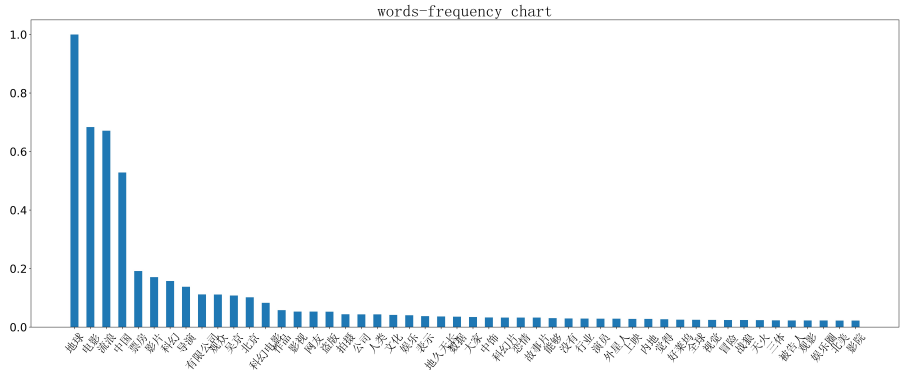

制作
中国
客观
观众
吴京
看过
电影院
没有
看到
地球
科幻片
木星
显示
好看
感觉
剧情
导演
演员
人类
美国
影片
科幻电影
加油
期待
宇宙
流浪
科幻
电影
希望
准备
喜欢
高兴
可能
知道
感动
时间
大家
孩子
震撼
战狼
票房
作品
小破球
看看
支持
国产
盗版
觉得
继续
值得

3. 女性用户


words-frequency chart
1.0
0.8
0.6
0.4
0.2
0.0



- 新闻



> 各个平台频度高的词有部分重复，但流浪、地球、电影、中国四个词在各大平台都是最高的。由高频的词汇可以看出各个平台评论的重点不同，例如豆瓣，侧重剧情、故事内容，新闻更多提到票房、导演、公司，而微博评论中提到较多的是人们的感受：好看、支持等以及演员、导演。

5.2 电影上映前后事件脉络（部分截图）

./result/result_news/sorted_top_50.txt保存了按时间排序的最重要的50篇新闻

```
#2018年07月09日 21-38 #《流浪地球》再曝贴片预告 吴京首演宇航员助力中国科幻.txt  0.5723767842362764
#2018年07月16日 02-39 #新京报 #新京报 #吴京上太空，科幻片《流浪地球》偏写实.txt 0.6413685862313556
#2018年11月14日 10-18 #电影《流浪地球》首曝科幻特辑 刘慈欣出镜深感拍摄不易.txt  0.6295980369633953
#2018年12月11日 22-33 #新浪网 #Daan-达岸电影 #《流浪地球》曝“行星发动机全景版”海报 放大数倍看满屏是细节.txt    0.5140445651467919
#2018年12月19日 10-23 #《流浪地球》发布“创想特辑”从无到有生动呈现刘慈欣的科幻想象.txt    0.5669382235197763
#2018年12月25日 07-03 #新浪网 #Daan-达岸电影 #《流浪地球》全国高校路演“冒险启程” 刘慈欣：影片充满质感.txt 0.5876693611798508
#2018年12月26日 10-26 #新浪网 #Daan-达岸电影 #《流浪地球》高校路演火热进行 吴孟达：郭帆导演是“疯子导演”.txt    0.59955022389543
#2018年12月28日 10-24 #新浪网 #Daan-达岸电影 #《流浪地球》高校路演火热进行 粉丝自制混剪视频吴孟达感动落泪.txt  0.5108827927448903
#2019年01月03日 23-10 #电影《流浪地球》全国高校路演完美收官 数万学生分享倔强故事.txt 0.5195930766733133
#2019年01月08日 10-49 #《流浪地球》曝意义特辑冒险小分队经受冰与火双重考验.txt    0.5632083493163854
#2019年01月21日 07-07 #新浪网 #Daan-达岸电影 #《流浪地球》举办超前点映 刘慈欣含泪“中国科幻终于启航”.txt    0.5978633986597812
#2019年01月27日 17-29 #新浪网 #第一制片人杂志 #这一次，《流浪地球》让我们终于有了可以抗衡好莱坞工业的电影.txt    0.6289531073485343
#2019年01月28日 09-43 #为啥要“流浪地球”？郭帆：房价贵 得带房子离开.txt 0.8464081192626907
#2019年01月29日 10-53 #新浪网 #Daan-达岸电影 #《流浪地球》首映口碑爆棚 观众笑称导演郭帆“空手套战狼”吴京.txt    0.6553450585898422
#2019年01月30日 11-28 #戴锦华评《流浪地球》：这是个“愚公移山”的故事.txt    0.6011000644220854
#2019年02月02日 12-47 #新浪网 #壹条电影 #《流浪地球》点映竟惹哭全场直男！中国有拿得出手的科幻电影了.txt    0.5554749003600432
#2019年02月02日 23-37 #《流浪地球》发布想象力主题海报 愿在孩子心中埋下想象力的种子.txt    0.5240180389124912
#2019年02月03日 09-11 #郭帆：拍摄《流浪地球》遇到的最大困难是信任.txt  0.6190896906878304
#2019年02月03日 11-30 #新浪网 #界面 #【专访】《流浪地球》导演郭帆：我们的科幻片还是小学生，一片蛮荒有待耕耘.txt 0.872158953457693
#2019年02月06日 00-40 #《流浪地球》 中国科幻片的种子终于发芽.txt  0.5337244116870755
#2019年02月06日 02-30 #评《流浪地球》：中国科幻片的种子终于发芽.txt    0.5408924298183206
#2019年02月06日 20-16 #《流浪地球》口碑领跑 吴京“带资进组”拼命出演.txt 0.6915791606272896
#2019年02月08日 16-26 #刘慈欣和西方科幻片给《流浪地球》带来了什么？.txt    0.6321129289969777
#2019年02月08日 17-03 #新浪娱乐 #新浪娱乐 #关于《流浪地球》，你需要知道的50件幕后故事.txt    0.9556196658708379
#2019年02月10日 02-06 #导演郭帆-拍摄《流浪地球》遇到的最大困难是信任.txt    0.614773562062862
#2019年02月10日 20-59 #刘慈欣回应《流浪地球》热点问题-没人能预测未来.txt    0.5195181703765932
#2019年02月11日 10-13 #刘慈欣：给《疯狂的外星人》《流浪地球》打满分.txt    0.5156698560314258
#2019年02月12日 01-58 #外国网友点赞《流浪地球》 有人花一天画吴京侧颜.txt    0.5818847211701647
#2019年02月12日 11-29 #新浪网 #十方娱乐2017 #电影《流浪地球》登陆二十亿电影俱乐部，中国科幻诚意启航.txt    0.6721434064583266
#2019年02月12日 15-56 #草根评《流浪地球》：视效震撼 大气磅礴.txt  1.0
#2019年02月17日 07-36 #《流浪地球》的资本游戏-阿里影业缘何能分一杯羹.txt    0.6832012110316502
#2019年02月19日 10-19 #新浪网 #Daan-达岸电影 #《流浪地球》导演郭帆“花样追星” 卡梅隆：中国电影应该有自信.txt    0.5503360054068056
#2019年02月20日 23-12 #新华社 #新华社-《流浪地球》开启中国科幻电影创作新征程.txt 0.5717064532661957
#2019年02月21日 09-53 #新浪娱乐 #新浪娱乐 #电影局局长：流浪地球成功得益于占据道义制高点.txt    0.5990637935036063
#2019年02月22日 10-44 #水墨风格塑造《流浪地球》浩瀚场面，中国科幻电影开启新征程.txt  0.5317573314353968
#2019年02月25日 09-53 #《流浪地球》 口碑高涨主创北美路演 海外华人祝福中国科幻.txt    0.5310399549935679
#2019年02月26日 10-38 #新浪网 #Daan-达岸电影 #《流浪地球》北美路演完美收官 郭帆导演：文化的启迪没有界限.txt    0.5363803270232719
#2019年02月27日 01-55 #《流浪地球》上焦点访谈 主创揭秘幕后制作过程.txt 0.6463197311454796
#2019年03月01日 10-45 #《流浪地球》发布视效特辑 郭帆：我希望让观众看到我们的成长.txt 0.5196330339586485
#2019年04月27日 11-18 #北京青年报 #北京青年报 #《流浪地球》遭高清盗版 中央领导做出重要批示.txt    0.5116221751620725
#2019年11月24日 06-08 #新浪网 #中国新闻网 #谁能成为下一个《流浪地球》？这些科幻IP有话说.txt  0.5119932485665287
#2019年11月25日 15-11 #谁能成为下一个《流浪地球》？这些科幻IP有话说.txt    0.5202110364778043
#2019年11月28日 12-05 #中国电影需要《流浪地球》更需要《红花绿叶》.txt  0.5156644011692838
#2019年12月07日 21-00 #新浪网 #传媒大眼 #2019内地票房破600亿：《哪吒》《流浪地球》《夏联4》前三.txt   0.5546025746829039
#2020年01月11日 20-25 #新浪网 #新京报 #郭帆透露正在打造《流浪地球2》，预计起码需要四年时间.txt    0.5159033155841832
#2020年01月11日 21-02 #新京报 #新京报 #郭帆透露正打造《流浪地球2》 预计起码需要四年.txt 0.5136770241341599
#2020年02月23日 22-07 #新浪网 #皮哥orange #当媒体夸赞《流浪地球》打开了中国科幻片大门时，黄建新笑而不语.txt    0.5124537857233235
#2020年03月18日 08-03 #新浪网 #1905电影网官博 #看·电影丨再看《流浪地球》，或许你能看懂更多.txt    0.5111134570874484
#2020年05月24日 21-30 #新浪网 #中国电影报道 #不止于《流浪地球》！这里堪称中国影视精品云集之地.txt 0.5367918395707119
#2020年05月25日 08-05 #新浪网 #1905电影网官博 #不止《流浪地球》！这里堪称中国影视精品云集地.txt  0.5369043094528421
```

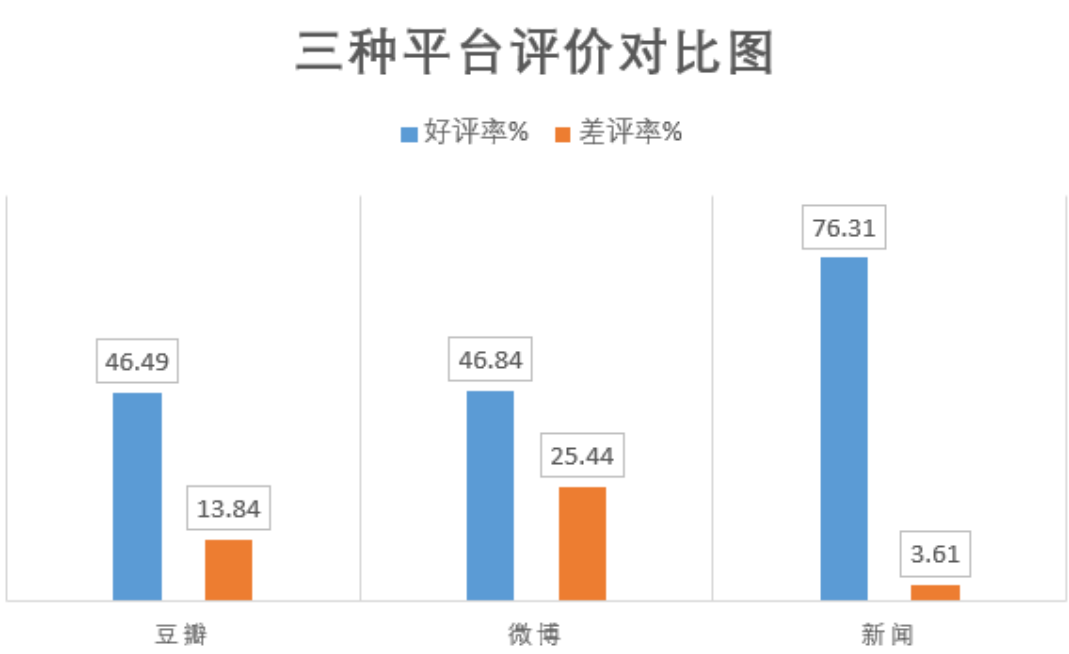
> 同时./result/result_news/abstact_news.txt保存了所有新闻的摘要内容。
> 由排序后的新闻可以了解从电影前至电影上映时至电影上映之后各大新闻媒体对该电影的报道，了解前后的全部事件脉络。新闻内容包括从上映前的预告，演员导演原著作者的宣传采访、上映时带来的反响变化以及之后关于下一步的筹备等等。

5.3大众评论情感分析

- 数据概述

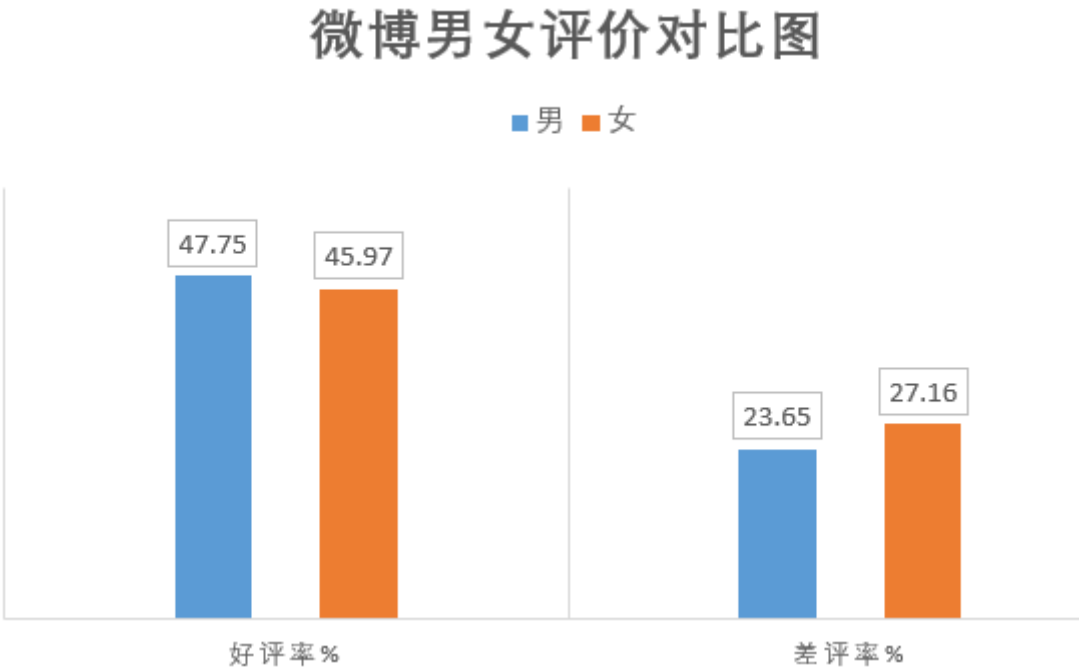
	豆瓣	微博	新闻
好评率%	46.49	46.84	76.31
差评率%	13.84	25.44	3.61
总数	19524	6176	249
好评数	9076	2893	190
差评数	2703	1571	9
	男	女	
好评率%	47.75	45.97	
差评率%	23.65	27.16	
总数	3028	3148	
好评数	1446	1447	
差评数	716	855	

- 平台评价对比图



> 由平台评价对比图可以看出豆瓣和微博的好评率差不多，都在46%左右，而相比于豆瓣，微博的差评率较高,几乎是豆瓣差评率的一倍。而新闻中好评率最高，差评率最低，仅3.61%。

- 微博男女评价对比图



> 根据微博男女评价对比图分析可得男性用户对该电影的认可度更高，相比于女性用户，男性用户对该电影的好评率略高，差评率较低。可以看出此类型电影更受男性观众的欢迎。

5.4频繁模式挖掘（部分截图）

由于相同词语的频率过低，因此，在进行频繁模式挖掘时将支持度设置为0.005,置信度为0.1。
关联规则：

- 豆瓣

```
180  ============================================================
181  frequent 3-itemsets     support
182  ============================================================
183  frozenset({'中国', '电影', '科幻电影'}) 0.005326777299733661
184  frozenset({'中国', '流浪', '地球'}) 0.00911698422454415
185  frozenset({'地球', '流浪', '电影'}) 0.015826674861708665
186  frozenset({'中国', '电影', '地球'}) 0.0059926244622003685
187  frozenset({'地球', '流浪', '科幻'}) 0.005121901249743905
188  ============================================================
189  Rules
190  ============================================================
191  frozenset({'科幻'})===>frozenset({'电影'})  conf: 0.19848975188781012
192  frozenset({'拯救'})===>frozenset({'地球'})  conf: 0.42023346303501946
193  frozenset({'科幻片'})===>frozenset({'流浪'})  conf: 0.11450381679389311
194  frozenset({'没有'})===>frozenset({'电影'})  conf: 0.21805792163543442
195  frozenset({'看到'})===>frozenset({'电影'})  conf: 0.2292134831460674
196  frozenset({'原著'})===>frozenset({'电影'})  conf: 0.2902097902097902
197  frozenset({'中国'})===>frozenset({'流浪'})  conf: 0.11167711598746083
198  frozenset({'流浪'})===>frozenset({'中国'})  conf: 0.12066045723962744
199  frozenset({'科幻'})===>frozenset({'地球'})  conf: 0.1790722761596548
200  frozenset({'电影'})===>frozenset({'地球'})  conf: 0.18996105581999134
201  frozenset({'地球'})===>frozenset({'电影'})  conf: 0.19909297052154196
202  frozenset({'太阳'})===>frozenset({'地球'})  conf: 0.5846153846153845
```

- 微博

```
 81  ============================================================
 82  frequent 3-itemsets     support
 83  ============================================================
 84  frozenset({'width', 'default', 'icon'}) 0.007448186528497409
 85  frozenset({'width', 'doge', 'icon'}) 0.00858160621761658
 86  frozenset({'good', 'width', 'icon'}) 0.00858160621761658
 87  frozenset({'width', 'url', 'icon'}) 0.01602979274611399
 88  frozenset({'地球', '流浪', '电影'}) 0.005019430051813471
 89  frozenset({'width', 'png', 'icon'}) 0.03222150259067358
 90  frozenset({'width', 'spanclass', 'icon'}) 0.005343264248704663
 91  frozenset({'url', 'spanclass', 'icon'}) 0.011981865284974092
 92  ============================================================
 93  Rules
 94  ============================================================
 95  frozenset({'url'})===>frozenset({'icon'})  conf: 1.0
 96  frozenset({'icon'})===>frozenset({'url'})  conf: 0.2252525252525253
 97  frozenset({'png'})===>frozenset({'width'})  conf: 0.7834645669291339
 98  frozenset({'width'})===>frozenset({'png'})  conf: 0.29879879879879884
 99  frozenset({'壁纸'})===>frozenset({'孟美岐'})  conf: 0.9756097560975608
100  frozenset({'孟美岐'})===>frozenset({'壁纸'})  conf: 0.18018018018018014
101  frozenset({'url'})===>frozenset({'width'})  conf: 0.4439461883408072
102  frozenset({'width'})===>frozenset({'url'})  conf: 0.14864864864864866
103  frozenset({'url'})===>frozenset({'spanclass'})  conf: 0.3318385650224215
104  frozenset({'spanclass'})===>frozenset({'url'})  conf: 0.40659340659340654
105  frozenset({'人类'})===>frozenset({'地球'})  conf: 0.5
106  frozenset({'电影'})===>frozenset({'流浪'})  conf: 0.10035842293906809
```

- 新闻

```
frozenset({'地球', '科幻', '中国', '传统', '电影'}) 0.006993006993006993
frozenset({'地球', '军号', '地久天长', '故事片', '中饰'}) 0.01048951048951049
frozenset({'地球', '吴京', '推介会', '电影', '喜欢'}) 0.006993006993006993
frozenset({'地球', '流浪', '吴京', '中国', '电影'}) 0.006993006993006993
frozenset({'地球', '流浪', '上榜', '郭帆', '导演'}) 0.01048951048951049
frozenset({'科幻', '科幻世界', '中国', '日本', '三体'}) 0.006993006993006993
frozenset({'地久天长', '金鸡奖', '故事片', '中饰', '电影'}) 0.006993006993006
frozenset({'地球', '流浪', '中国', '作品', '电影'}) 0.006993006993006993
frozenset({'地球', '流浪', '中国', '郭帆', '电影'}) 0.01048951048951049
frozenset({'地球', '中国', '北京', '有限公司', '电影'}) 0.013986013986013986
frozenset({'林超贤', '地久天长', '故事片', '中饰', '电影'}) 0.006993006993006
frozenset({'地球', '流浪', '中国', '刘慈欣', '电影'}) 0.006993006993006993
frozenset({'地球', '流浪', '票房', '战狼', '电影'}) 0.006993006993006993
frozenset({'地球', '流浪', '中国', '郭帆', '观众'}) 0.006993006993006993
frozenset({'地球', '科幻', '中国', '郭帆', '观众'}) 0.006993006993006993
frozenset({'焦裕禄', '文化', '影视', '山东', '电影'}) 0.006993006993006993
frozenset({'地球', '流浪', '中国', '开启', '科幻电影'}) 0.006993006993006993
frozenset({'地球', '吴京', '郭帆', '拍摄', '导演'}) 0.006993006993006993
frozenset({'地球', '流浪', '中国', '电影', '观众'}) 0.006993006993006993
frozenset({'地球', '流浪', '科幻', '中国', '电影'}) 0.01048951048951049
frozenset({'地球', '流浪', '新片', '角逐', '天坛'}) 0.006993006993006993
==========================================================
Rules
==========================================================
frozenset({'继续'})===>frozenset({'地球'})  conf: 1.0
frozenset({'外星人'})===>frozenset({'流浪'})  conf: 0.39999999999999997
frozenset({'焦裕禄'})===>frozenset({'山东'})  conf: 1.0
frozenset({'山东'})===>frozenset({'焦裕禄'})  conf: 1.0
frozenset({'观众'})===>frozenset({'佳音'})  conf: 0.10526315789473685
frozenset({'佳音'})===>frozenset({'观众'})  conf: 0.6666666666666666
frozenset({'导演'})===>frozenset({'影片'})  conf: 0.125
frozenset({'影片'})===>frozenset({'导演'})  conf: 0.11111111111111112
frozenset({'没有'})===>frozenset({'郭帆'})  conf: 0.5
frozenset({'焦裕禄'})===>frozenset({'影视'})  conf: 1.0
frozenset({'影视'})===>frozenset({'焦裕禄'})  conf: 0.2857142857142857
frozenset({'地久天长'})===>frozenset({'电影'})  conf: 0.5714285714285714
frozenset({'奖项'})===>frozenset({'电影'})  conf: 1.0
frozenset({'票房'})===>frozenset({'春节'})  conf: 0.11428571428571428
frozenset({'春节'})===>frozenset({'票房'})  conf: 0.5714285714285714
frozenset({'希望'})===>frozenset({'电影'})  conf: 1.0
```

6. 任务分配

- 项目设计: 吴嘉豪 、刘金田
- 数据收集: 吴嘉豪、刘金田、孙玥
- 数据处理：刘金田、易鑫
- 算法实现: 刘金田、易鑫、吴嘉豪、李祥潮
- 可视化:刘金田、李祥潮
- 文档编写:孙玥、吴嘉豪、李祥潮