# B站弹幕数据分析

## 背景介绍

能够在观看视频的过程中发表自己的评论，并且评论可以在你所希望的时间点、位置以滑行或停留的方式出现在视频中，所有观看视频的人都可以看见评论，这样一类的评论叫做弹幕，此类网站叫弹幕网站。

在视频软件层出不穷的当下，除了优质的视频内容，对于社区文化如何探索成为当下各个公司极力探索的目标。新时期自媒体盛行，BiliBili（即B站）相较于同类软件，其社区文化呈现欣欣向荣之势，拥有较高的日活跃量。而弹幕文化俨然成为其不可或缺的一部分。

通过对于当前时间热门视频中弹幕进行分析，将数据进行可视化处理，既可以知道在这一时间段网络舆论流行的大体趋势，把握用户心理态度，加强受众的互动反馈，还可以激发用户对于弹幕文化的探索的兴趣，使得用户保持新鲜度，延长软件寿命。

## 数据来源

此次分析的数据为B站热门视频【2022十大影视文艺复兴】全是名场面，快乐源泉 (<https://www.bilibili.com/video/BV1X24y1D7LW/>)的弹幕，利用python爬虫技术爬取该视频的弹幕数据并存储到csv文件中。

爬虫代码如下：

1. import re
2. import requests
3. url = f'https://comment.bilibili.com/926923202.xml'
4. res = requests.get(url)
5. with open(f'926923202.xml', 'wb') as f:
6. f.write(res.content)
7. with open('926923202.xml', encoding='utf-8') as f:
8. data = f.read()
9. comments = re.findall('<d p="(.\*?)">(.\*?)</d>', data)
10. # print(len(comments))  # 3000
11. danmus = [','.join(item) for item in comments]
12. headers = ['stime', 'mode', 'size', 'color', 'date', 'pool', 'author', 'dbid', 'text']
13. headers = ','.join(headers)
14. danmus.insert(0, headers)
15. with open('danmus.csv', 'w', encoding='utf\_8\_sig') as f:
16. f.writelines([line+'\n' for line in danmus])

得到的数据如下图所示：

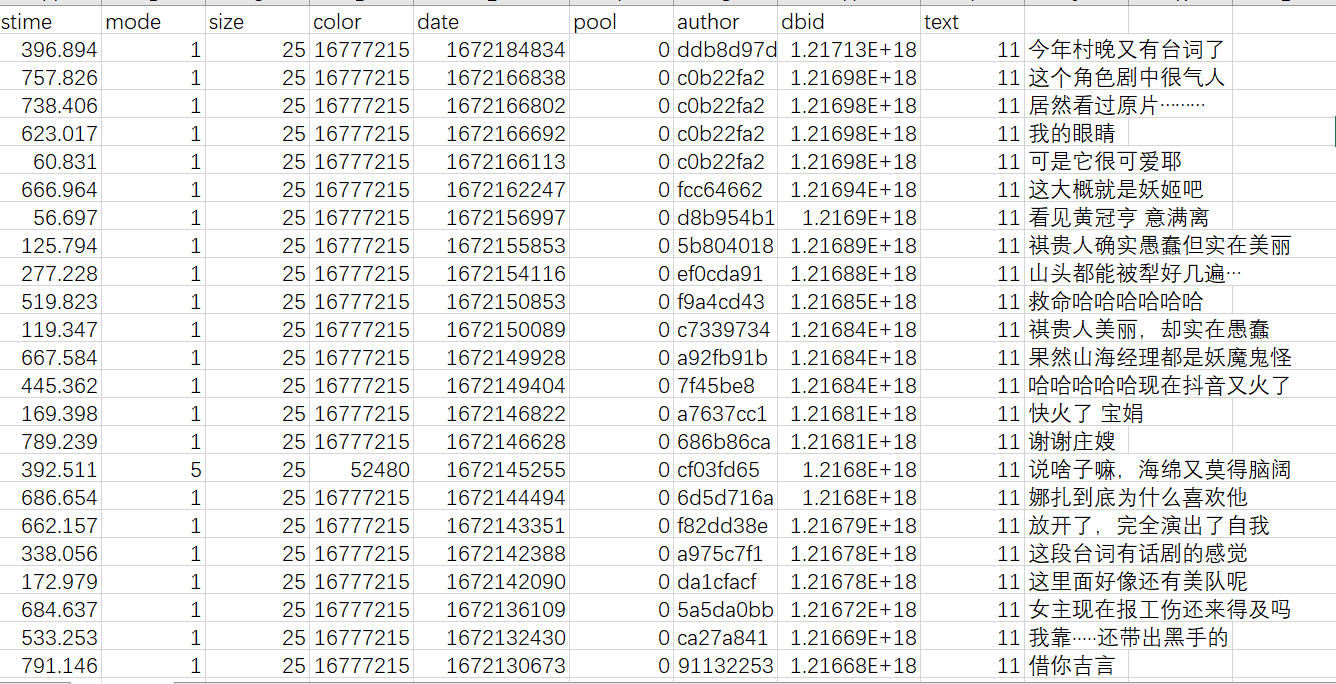


图2.1得到的数据展示

## 数据分析

**1.词云**

利用python的jieba库对弹幕文本进行分词处理，然后利用WordCloud库生成词云，并利用pyecharts进行可视化。代码如下：

1. from pyecharts import options as opts
2. from pyecharts.charts import WordCloud
3. import jieba
4. with open('danmus.csv', encoding='utf-8') as f:
5. text = " ".join([line.split(',')[-1] for line in f.readlines()])
6. words = jieba.cut(text)
7. \_dict = {}
8. for word in words:
9. if len(word) >= 2:
10. \_dict[word] = \_dict.get(word, 0)+1
11. items = list(\_dict.items())
12. items.sort(key=lambda x: x[1], reverse=True)
13. c = (
14. WordCloud()
15. .add(
16. "",
17. items,
18. word\_size\_range=[20, 120],
19. textstyle\_opts=opts.TextStyleOpts(font\_family="cursive"),
20. )
21. .render("wordcloud.html")
22. )

生成的词云如下图所示：



图3.1 词云

通过上图可以看出，“哈哈哈”、“谢谢”、“真的”等词在弹幕中出现的频率较高。

**2.情感分析**

利用snownlp库对弹幕文本进行一个简单的三分类的情感分析，挖掘弹幕反映的用户情感色彩。代码如下：

1. from snownlp import SnowNLP
2. from pyecharts import options as opts
3. from pyecharts.charts import Pie
4. with open('danmus.csv', encoding='utf-8') as f:
5. text = [line.split(',')[-1] for line in f.readlines()[1:]]
6. emotions = {
7. 'positive': 0,
8. 'negative': 0,
9. 'neutral': 0
10. }
11. for item in text:
12. if SnowNLP(item).sentiments > 0.6:
13. emotions['positive'] += 1
14. elif SnowNLP(item).sentiments < 0.4:
15. emotions['negative'] += 1
16. else:
17. emotions['neutral'] += 1
18. print(emotions)
19. c = (
20. Pie()
21. .add("", list(emotions.items()))
22. .set\_colors(["blue", "purple", "orange"])
23. .set\_series\_opts(label\_opts=opts.LabelOpts(formatter="{b}: {c} ({d}%)"))
24. .render("emotionAnalysis.html")
25. )

情感分析的结果如下图所示：

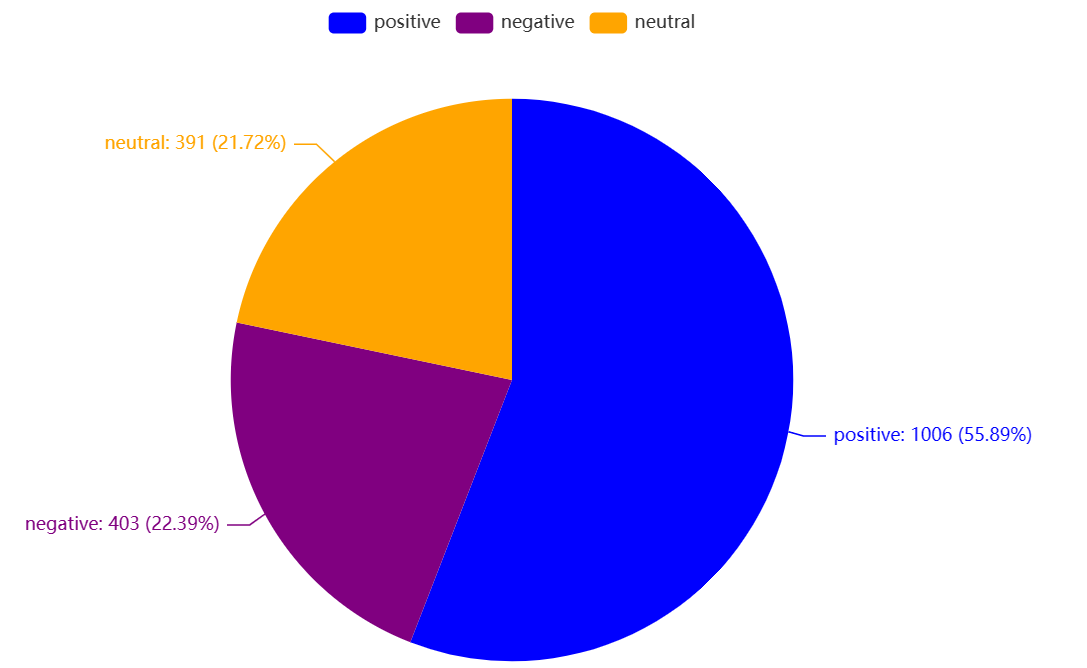


图3.2 情感分析结果

通过上图可以发现，弹幕文本反映的情感倾向大都是positive，即积极乐观的，说明大多数观众对于该视频是喜欢的。

**3.** **精彩片段**

通过对不同时间段弹幕数量的统计，并绘制折线图，可以发现哪些时间段的弹幕数量较多，从而找到该视频的精彩片段。代码如下：

1. from pyecharts.commons.utils import JsCode
2. from pyecharts.charts import Line
3. from pyecharts.charts import Line, Grid
4. import pyecharts.options as opts
5. with open('danmus.csv', encoding='utf-8') as f:
6. text = [float(line.split(',')[0]) for line in f.readlines()[1:]]
7. text = sorted([int(item) for item in text])
8. data = {}
9. for item in text:
10. item = int(item/60)
11. data[item] = data.get(item, 0)+1
12. x\_data = list(data.keys())
13. y\_data = list(data.values())
14. background\_color\_js = (
15. "new echarts.graphic.LinearGradient(0, 0, 0, 1, "
16. "[{offset: 0, color: '#c86589'}, {offset: 1, color: '#06a7ff'}], false)"
17. )
18. area\_color\_js = (
19. "new echarts.graphic.LinearGradient(0, 0, 0, 1, "
20. "[{offset: 0, color: '#eb64fb'}, {offset: 1, color: '#3fbbff0d'}], false)"
21. )
22. c = (
23. Line(init\_opts=opts.InitOpts(bg\_color=JsCode(background\_color\_js)))
24. .add\_xaxis(xaxis\_data=x\_data)
25. .add\_yaxis(
26. series\_name="弹幕数量",
27. y\_axis=y\_data,
28. is\_smooth=True,
29. symbol="circle",
30. symbol\_size=6,
31. linestyle\_opts=opts.LineStyleOpts(color="#fff"),
32. label\_opts=opts.LabelOpts(is\_show=True, position="top", color="white"),
33. itemstyle\_opts=opts.ItemStyleOpts(
34. color="red", border\_color="#fff", border\_width=3
35. ),
36. tooltip\_opts=opts.TooltipOpts(is\_show=True),
37. areastyle\_opts=opts.AreaStyleOpts(
38. color=JsCode(area\_color\_js), opacity=1),
39. markpoint\_opts=opts.MarkPointOpts(
40. data=[opts.MarkPointItem(type\_="max")])
41. )
42. .set\_global\_opts(
43. title\_opts=opts.TitleOpts(
44. title="",
45. pos\_bottom="5%",
46. pos\_left="center",
47. title\_textstyle\_opts=opts.TextStyleOpts(
48. color="#fff", font\_size=16),
49. ),
50. xaxis\_opts=opts.AxisOpts(
51. type\_="category",
52. boundary\_gap=False,
53. axislabel\_opts=opts.LabelOpts(margin=30, color="#ffffff63"),
54. axisline\_opts=opts.AxisLineOpts(
55. linestyle\_opts=opts.LineStyleOpts(width=2, color="#fff")
56. ),
57. axistick\_opts=opts.AxisTickOpts(
58. is\_show=True,
59. length=25,
60. linestyle\_opts=opts.LineStyleOpts(color="#ffffff1f"),
61. ),
62. splitline\_opts=opts.SplitLineOpts(
63. is\_show=True, linestyle\_opts=opts.LineStyleOpts(color="#ffffff1f")
64. )
65. ),
66. yaxis\_opts=opts.AxisOpts(
67. type\_="value",
68. position="left",
69. axislabel\_opts=opts.LabelOpts(margin=20, color="#ffffff63"),
70. axisline\_opts=opts.AxisLineOpts(
71. linestyle\_opts=opts.LineStyleOpts(width=2, color="#fff")
72. ),
73. axistick\_opts=opts.AxisTickOpts(
74. is\_show=True,
75. length=15,
76. linestyle\_opts=opts.LineStyleOpts(color="#ffffff1f"),
77. ),
78. splitline\_opts=opts.SplitLineOpts(
79. is\_show=True, linestyle\_opts=opts.LineStyleOpts(color="#ffffff1f")
80. ),
81. ),
82. legend\_opts=opts.LegendOpts(is\_show=False),
83. tooltip\_opts=opts.TooltipOpts(trigger="axis", axis\_pointer\_type="line")
84. )
85. .render("highlights.html")
86. )

结果如下图所示：

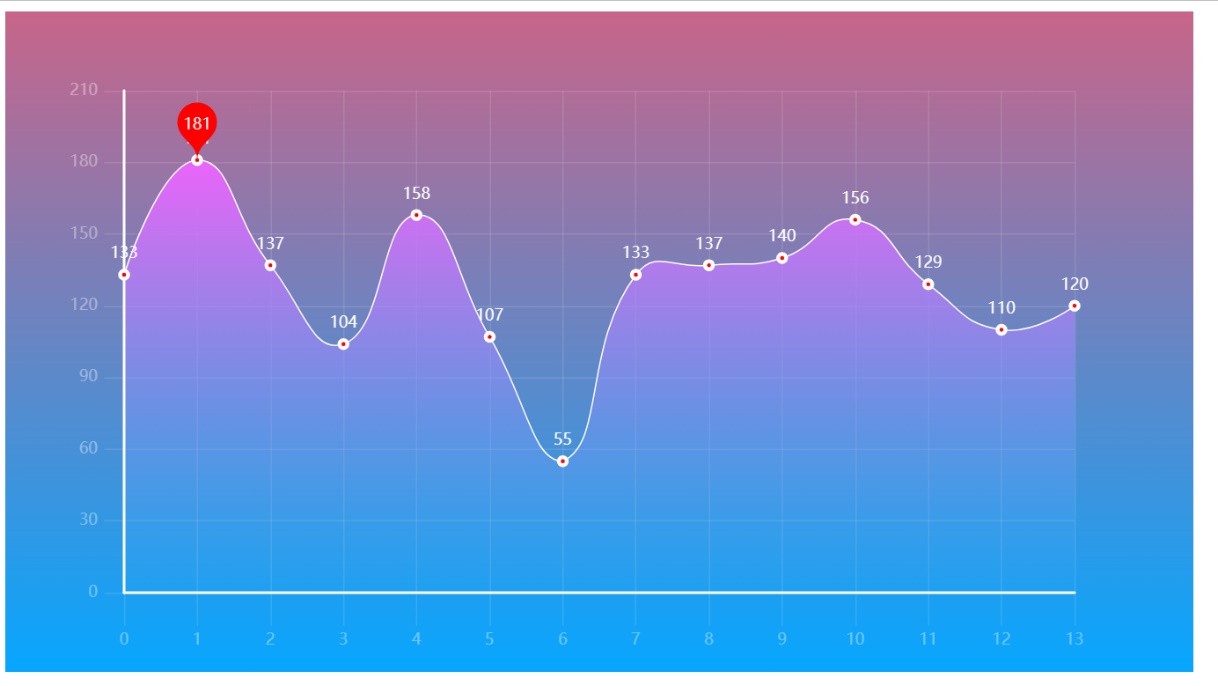


图3.3弹幕数量统计折线图

通过上图可以发现，该视频的第1分钟、第4分钟以及第10分钟弹幕数量较多，可能为该视频的精彩片段。

## 结论

通过对该视频的弹幕进行词云统计、情感分析、精彩片段分析，可以得出结论，该视频弹幕的高频词汇是“哈哈哈”、“谢谢”、“真的”等词，观众的主要情感倾向都是积极的，说明观众对于该视频持喜爱态度，该视频的精彩片段在第1分钟、第4分钟以及第10分钟。