基于文本挖掘的观众对国产电影的比较分析

摘要：本文以国产电影《流浪地球 2》影评为科幻电影研究对象，分别从词频、主题分析、情感分析三方面内容进行比较分析，旨在挖掘出影片的特点、受众对电影的反响和口碑等信息。对电影制作方而言，使用文本挖掘技术能够更好地了解观众需求和市场趋势，进而提升电影口碑和票房。同时，这一技术也能够为影评人和观众们提供更准确的反馈和评价。为未来国产电影科幻题材具有研究意义。

关键词：国产电影；科幻；《流浪地球 2》；影评

1.引言

随着国内电影市场的快速发展，国产电影也呈现出了多元化与个性化的发展趋势，对于这些国产电影的评价与观众反馈成为了研究的热点。在这个背景下，文本挖掘技术为我们提供了一个强大的工具，能够帮助我们从大量文本数据中挖掘出影片的特点和受众的反馈。也是一种通过计算机技术来对大量的文本数据进行分析、提取信息的技术。在电影市场中，通过对电影评论、社交媒体等文本数据进行分析，可以挖掘出影片的特点、受众对电影的反响和口碑等信息。这对于电影制作方来说，能够帮助他们更好地掌握观众需求和市场趋势，提高电影口碑和票房。而对于电影评论者和观众来说，也能够提供更准确的反馈和评价。

2.研究对象及方法

本文采用Python文本挖掘，对《流浪地球 2》的电影影评进行爬取，并对观众对这部电影的评价做了深入的分析。

这部电影不仅讲述了全人类面临的危机，同时也展现了人类坚韧不拔、勇往直前的精神。在电影中，我们看到了航天员们用自己的生命换取了人类的安全，看到了科学家们的聪明才智和顽强求知的精神。同时，在面临这个未知的危机时，我们也看到了人与人之间的紧密联系，包括家人、爱人、朋友和同事等之间的互相支持和守护。通过这些元素的融合，这部电影让我们更加深刻地认识到了人类的伟大和生命的可贵，也让我们更加珍惜我们身边的每一个人和我们共同生存的这个星球。

本文利用文本挖掘技术对观众对国产电影的评价和反馈进行分析，并试图寻找出影响国产电影口碑的主要因素，以期为国产电影的发展提供参考和建议。本文以观众在豆瓣电影网站上的影评为样本，通过对影评进行情感分析、主题提取、

关键词提取等文本挖掘方法，来揭示影片的主题、评价和受众反应，以此探讨国产电影发展的现状和趋势。

以关键词“流浪地球 2”搜索电影影评，共爬取到了 595 篇影评内容，有效样本共计 590 篇，并基于上述文本进行文本挖掘。

3.研究过程及发现

3.1 国产电影《流浪地球 2》词频分析

- 在读入《流浪地球 2》影评数据后，首先针对缺失值和重复值进行查看，并有针对性的处理，便于后续进行清洗和分词等操作。在数据清洗时主要用到 re 函数对文本中的中文提取，去除了异常表情符号等。
- 在词频分析时采用 jieba 分词函数，应用 cut 精确切分模式对影评样本进行切词，并使用停用词表删除了无实际意义的冠词，虚词，拟声词等等。通过统计词语的数量，并降序排序，给出了词频统计的 top50，并绘制图形。

影评词频分析如下：

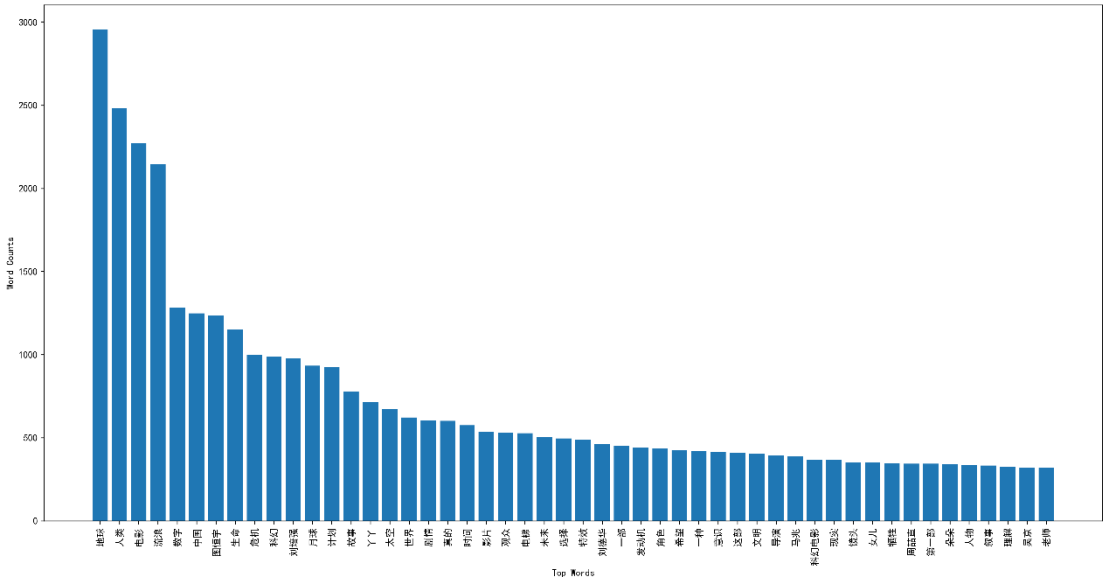


图 3-1 词频分析

词频往往可以清晰地勾勒出所要解读内容的重点并加以分析，将影评的词频统计降序排列后，可以看出“地球”“人类”“中国”“科幻”等词语占比较高。这说明这部电影与环保、人类命运和中国等议题有着紧密的关系。

现如今国产电影不断崛起，科幻题材系列往往与中国和地球以及人类家园题材联系，这一分析结果有助于电影在选材方面的考量，制作组往这一方向前进，

往往贴合观众的口碑。

3.2 国产电影《流浪地球 2》主题建模

国产电影影评的 LDA 主题分析是一种文本挖掘技术，可以对《流浪地球 2》中的评论文本进行主题提取和分析，从而了解用户的关注点和热点话题。LDA（Latent Dirichlet Allocation）是一种主题模型，可以将文本数据转化为主题-词语分布的概率模型，从而实现对文本主题的自动发现和分析。

3.2.1 主题分析流程

在进行 LDA 主题分析之前，需要对国产电影《流浪地球 2》中的评论文本进行数据预处理，包括删除无效字符、停用词过滤、语料转换等。然后，将预处理后的评论文本转化为词袋模型，并使用 LDA 模型对其进行主题分析。具体步骤如下：

1.构建词袋模型：将预处理后的评论文本转化为词袋模型，即将每个文本表示为一个向量，向量中的每个元素表示一个词语的出现次数。

2.训练 LDA 模型：使用训练集数据训练 LDA 模型，得到每个主题的词语分布和每个文本的主题分布。

3.选择主题数：根据主题的质量和数量选择最优的主题数，一般采用 perplexity 和 coherence 等指标评估主题的质量。

4.分析主题：对每个主题进行分析，了解主题的关键词、主题分布和主题含义，从而了解用户的关注点和热点话题。

3.2.2 一致性得分寻优曲线

一致性得分（coherence score）是一种评估 LDA 主题模型质量的指标，可以用于主题数。一致性得分越高，表示主题模型的质量越好，可以更好地反映文本数据中的主题结构。在确定影评主题数时，可以通过计算不同主题数下的一致性得分，选择一致性得分最高的主题数作为最终的主题数。

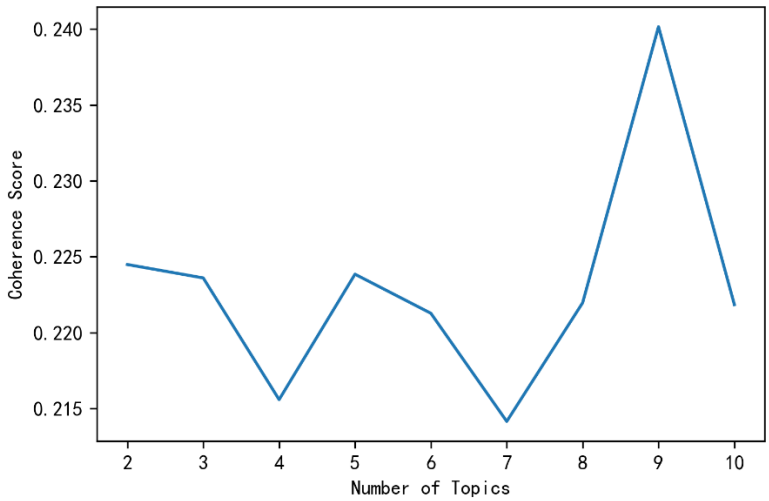


图 3-2 一致性得分寻优曲线

主题寻优曲线可以帮助我们确定最佳的主题数。在主题寻优曲线上，横轴表示主题数，纵轴表示主题一致性得分。一致性得分越高，表示主题之间的关联性越强，主题分析结果越好。

通常，可以选择一致性得分最高的主题数作为最佳主题数。但是，在实际应用中，最佳主题数还需考实际效果。如果主题数过少，可能会导致主题之间的关联性不够强，无法发现主题之间的细节和差异。如果主题数过多，可能会导致主题之间的关联性过于复杂，难以解释和理解。

3.2.3 主题提取实验结果分析

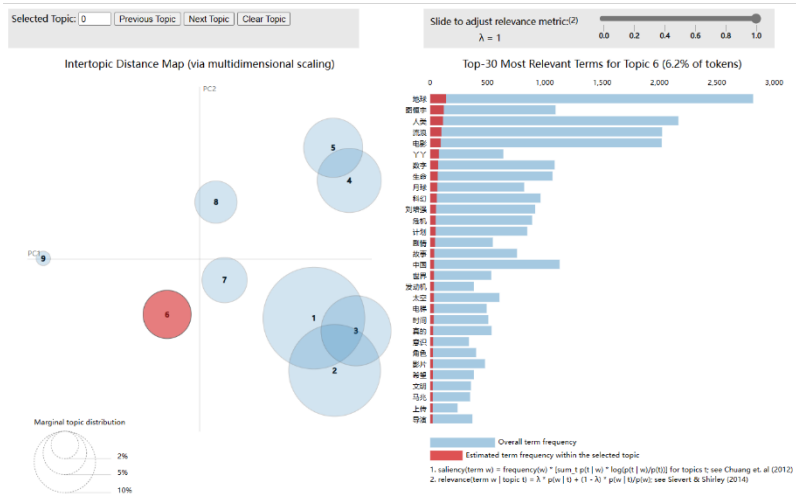
本文根据实验结果一致性得分最高处选取了 9 个主题，各主题词如下表

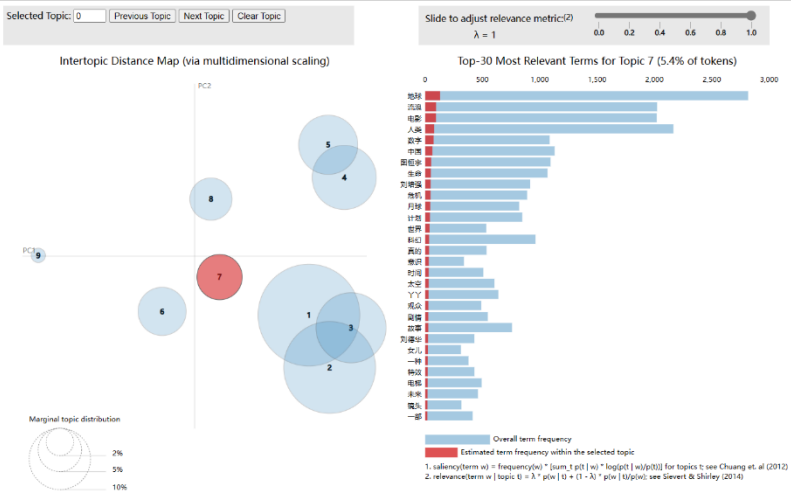
	主题词1	主题词2	主题词3	主题词4	主题词5	主题词6	主题词7	主题词8	主题词9	主题词10
主题1	地球	流浪	电影	人类	数字	中国	图恒宇	生命	刘培强	危机
主题2	地球	电影	流浪	人类	中国	刘培强	生命	图恒宇	计划	月球
主题3	人类	地球	数字	电影	流浪	生命	中国	危机	图恒宇	科幻
主题4	地球	电影	流浪	人类	生命	科幻	数字	故事	中国	刘培强
主题5	人类	电影	地球	流浪	数字	中国	危机	月球	图恒宇	科幻
主题6	地球	图恒宇	人类	流浪	电影	丫丫	数字	生命	月球	科幻
主题7	地球	流浪	人类	电影	图恒宇	生命	危机	科幻	中国	刘培强
主题8	人类	电影	流浪	中国	地球	危机	刘培强	计划	数字	生命
主题9	图恒宇	人类	地球	电影	流浪	数字	生命	月球	危机	刘培强

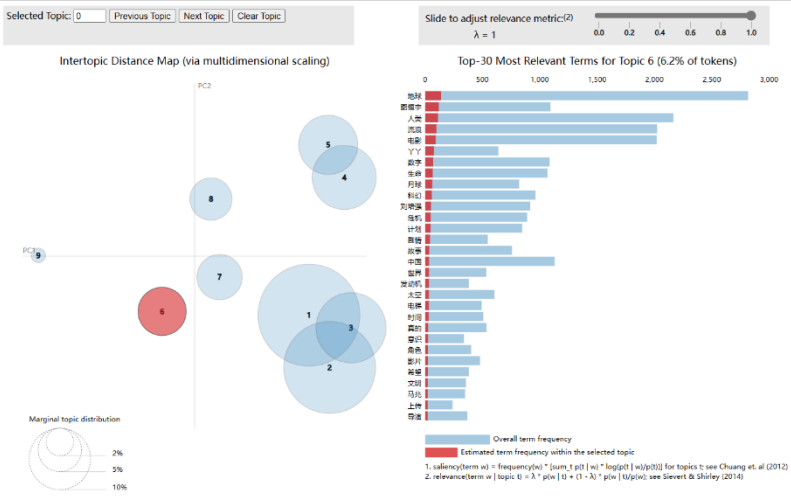
图 3-3 主题词分布

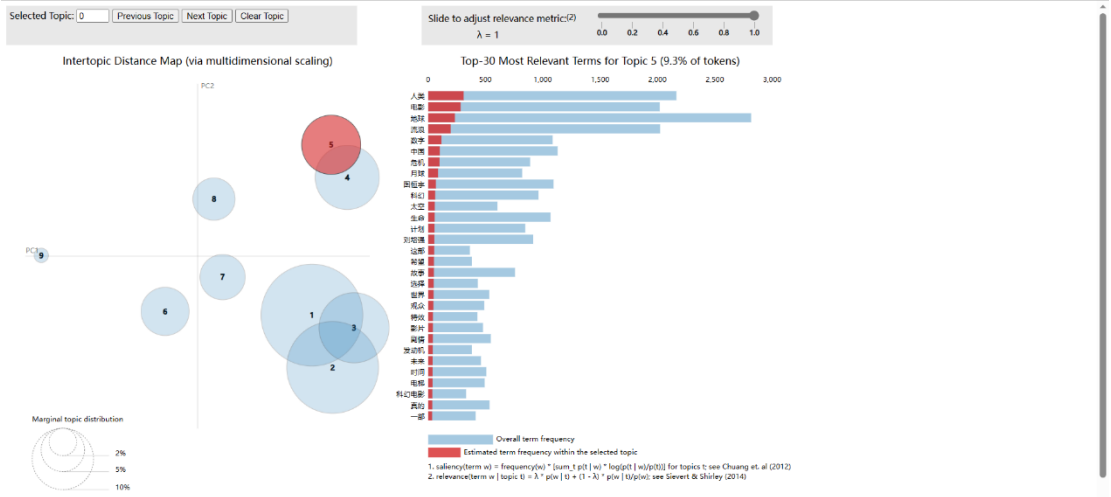
通过 LDA 主题分析，可以发现，《流浪地球 2》影评中用户的关注点主要集中在电影内容情节本身，以及对人类家园和地球未来的思索。

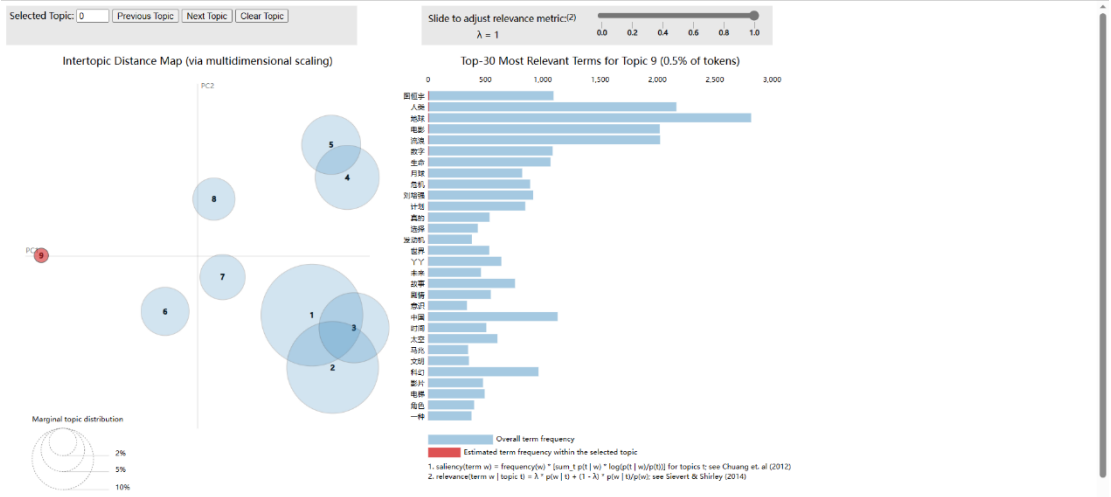
通过构造词典、去高频词、构造语料库后，训练了一个 LDA 模型。研究时选择了生成 9 个主题数，每个主题中包含 20 个词。接着使用pyLDAvis 对 LDA 主题模型进行可视化处理，结果如下：


Selected Topic: 0
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0 0.2 0.4 0.6 0.8 1.0
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 6 (6.2% of tokens)
0 500 1,000 1,500 2,000 2,500 3,000
地球
图根市
人类
流浪
电影
丫丫
数字
生命
月球
科幻
刘慈强
危机
计划
詹姆
故事
中国
世界
发动机
太空
电梯
时间
真的
登记
角色
影片
希望
文明
马兵
上帝
导弹
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)


Selected Topic: 0
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0 0.2 0.4 0.6 0.8 1.0
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 7 (5.4% of tokens)
0 500 1,000 1,500 2,000 2,500 3,000
地球
流浪
电影
人类
数字
中国
图根市
生命
刘慈强
危机
月球
计划
世界
科幻
真的
登记
时间
太空
丫丫
观众
詹姆
故事
刘慈欣
女儿
一种
特效
电梯
未来
镜头
一部
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)


Selected Topic: 0
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0 0.2 0.4 0.6 0.8 1.0
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 6 (6.2% of tokens)
0 500 1,000 1,500 2,000 2,500 3,000
地球
图根市
人类
流浪
电影
丫丫
数字
生命
月球
科幻
刘慈强
危机
计划
詹姆
故事
中国
世界
发动机
太空
电梯
时间
真的
登记
角色
影片
希望
文明
马兵
上帝
导弹
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)


Selected Topic: 0
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0 0.2 0.4 0.6 0.8 1.0
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 5 (9.3% of tokens)
0 500 1,000 1,500 2,000 2,500 3,000
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)


Selected Topic: 0
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0 0.2 0.4 0.6 0.8 1.0
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 9 (0.5% of tokens)
0 500 1,000 1,500 2,000 2,500 3,000
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

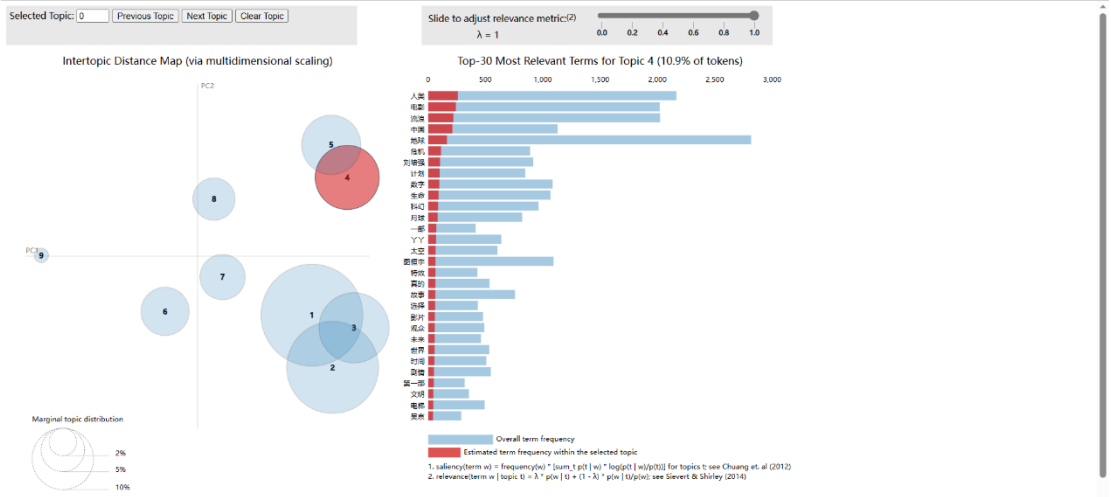

Selected Topic: 0
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0 0.2 0.4 0.6 0.8 1.0
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 4 (10.9% of tokens)
0 500 1,000 1,500 2,000 2,500 3,000
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

图 3-4 主题气泡图

左侧图中的每个气泡代表一个主题。气泡越大，该主题就越普遍。

一个好的主题模型将在整个图表中分散相当大的非重叠气泡，而不是聚集在一个象限中。

具有太多主题的模型通常会有许多重叠，小尺寸的气泡聚集在图表的一个区域中。

如果将光标移动到其中一个气泡上，右侧的单词和条形将会更新。这些单词是构成所选主题的显著关键字。

从主题关键词可以看出“数字中国”、“剧情”，“科幻”，“崛起”等主题词被显著提及，说明观众对于国产科幻电影充满了期待，对中国科幻电影的崛起抱有巨大希望！

这一结果也佐证了之前词频分析的国产电影口碑。

3.3 国产电影《流浪地球 2》情感倾向分析

在情感倾向分析中，由于评论是中文，本文采用 snownlp 情感分析工具库，对影评内容做出情感倾向评分，分析结果如下：

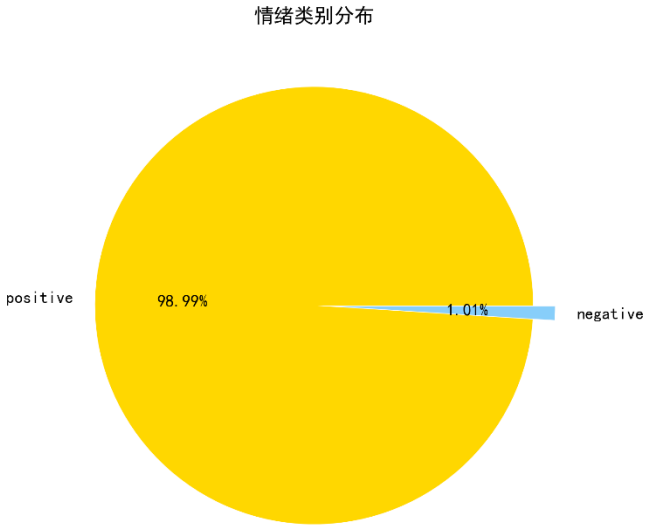


图 3-5 情感类别占比

Snownlp 情感评分的结果在 0-1 之间，值越大，说明情感倾向越积极，越小说明情感倾向越消极。本文以 0.5 为分界值，划分情感得分为积极与消极，并做出饼图统计积极与消极类别的占比。

从图中可以看出，积极影评占比超过 98%，进一步说明观众对国产电影科幻题材系列的高度赞扬。

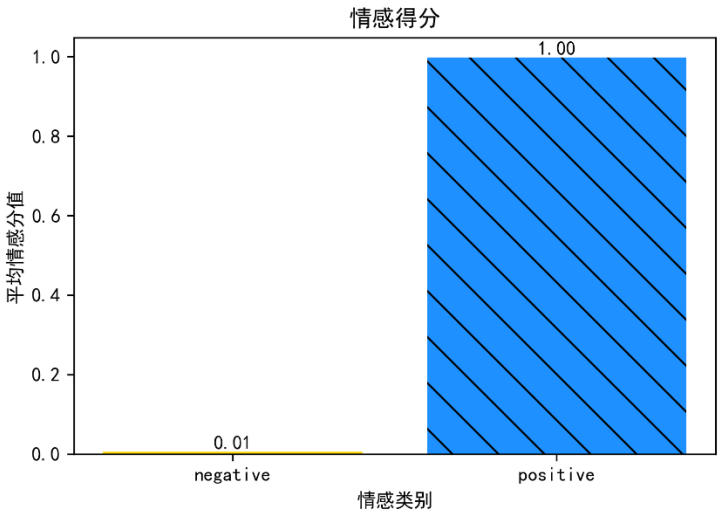


图 3-6 平均情感得分统计

通过对积极与消极情感得分的平均值统计，以积极消极类别为横坐标做出柱形图，可以发现，积极类别的平均情感得分接近为 1，消极情感得分平均为 0.01，说明在积极评价中，观众给出的积极情感比较强烈，当然也不排除一些极端的消极影评。

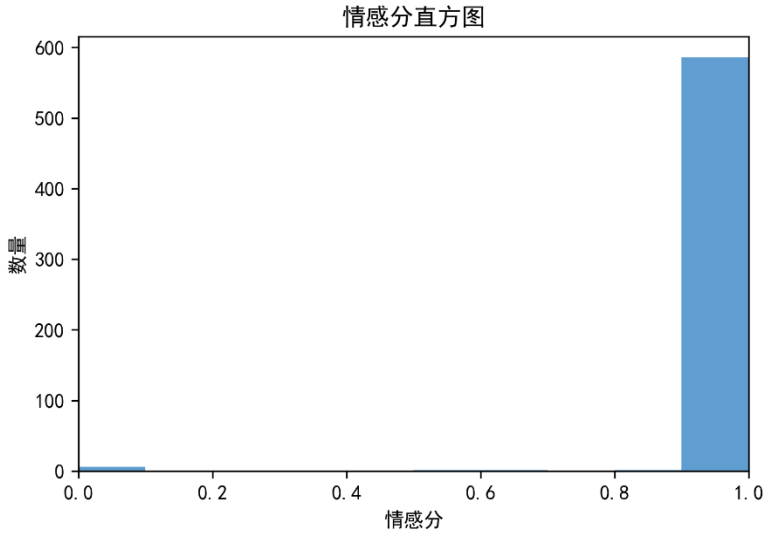


图 3-7 情感得分分布

通过每隔 0.1 区间，划分情感得分为 10 个小阶段，来统计下观众对电影情感的区间分布情况。

可以看到大部分影评情感得分在 0.9-1 区间，少部分分布在 0-0.1 之间。这一结果印证了上述平均情感得分的统计结果，说明观众对国产电影的科幻题材系列正向评价呼声较高。

图 3-8 词云图

从词云图中可以看出，“地球”,“人类”,“电影”,“数字中国”等词语出现频率较高，这表明观众对于国产电影科幻系列的关注度较高。另外,“故事”,“设定”等词语也出现频率较高，这表明观众对于电影剧情的需要也有所关注。

通过这些关键词的分析，可以深入了解观众对于科幻电影主题的关注点和态度，为相关决策和沟通提供参考。同时，也可以帮助制作组更好地了解电影剧情和公众的反应，及时采取措施应对。

3.4 国产电影 LSTM 情感倾向分类模型

国产电影 LSTM 情感倾向分类模型是基于深度学习中的长短期记忆网络（LSTM）框架构建而成的，主要用于对电影评论的情感倾向进行分类，包括正面、负面和中性三类。该模型通过对大规模情感标注的电影评论数据集进行训练，优化神经网络参数，使其能够自动识别和提取评论中的情感特征，并将其映射到对应的情感分类标签上，从而实现对电影评论情感倾向的智能分类判断。模型具有较高的准确率和稳定性，在情感倾向分类任务中得到了广泛应用。

数据预处理：

```
#定义分词函数，对传入的x进行分词
cw = lambda x: list(jieba.cut(str(x)))
# apply传入一个函数，把cw函数应用到data['review']的每一行
# 把分词后的结果保存到data['words']中
data['words'] = data['review'].apply(cw)
```

对数据集中的评论文本进行中文分词操作。代码中定义了一个 lambda 函数，命名为 cw，用于将传递给它的文本进行分词，函数使用 jieba 库中的 cut 方法分

词，并将分词结果转化为列表形式。

接下来，使用 Pandas 中的 apply 方法将 cw 函数应用到 data 数据集中的'review'列中的所有行上，得到一个新的'words'列，该列中每一个元素都是'review'中对应行的文本分词后的列表。最终的输出结果就是 data 数据集增加了一个'words'列，该列中每个元素都是对应"review"文本的分词结果。

```
# 实例化Tokenizer，设置字典中最大词汇数为30000
# Tokenizer会自动过滤掉一些符号比如：!"#$%&()*+,-./:;<=>?@[\]^_`{|}~\t\n
tokenizer = Tokenizer(num_words=30000)
# 传入我们的训练数据，建立词典，词的编号根据词频设定，频率越大，编号越小，
tokenizer.fit_on_texts(texts)
```

使用 Keras 中的 Tokenizer 类对给定的文本进行分词，并构建一个词典。具体而言，该类将所有文本中的单词按照词频（出现次数）进行排序，并为每个词汇赋予一个唯一的整数编号。Tokenizer 的 num_words 参数被设置为 30000，意味着该词典中最多只会包含 30000 个单词。

接下来，使用 fit_on_texts 方法将所有的训练数据进行分词和编号操作，以构建出该 Tokenizer 对象的内部词典。最终该 Tokenizer 对象存储了该数据集中所有单词的编号，以及每个单词的词频。

需要注意的是，Tokenizer 会自动过滤掉一些符号和空格等特殊字符，比如!"#$%&()*+,-./:;<=>?@[\]^_`{|}~\t\n 等。最终的输出结果是分词器 tokenizer，所有的数据都是基于这个 tokenizer 进行编码的。

```
# 把词转换为编号，编号大于30000的词会被过滤掉
sequences = tokenizer.texts_to_sequences(texts)
# 把序列设定为max_length的长度，超过max_length的部分舍弃，不到max_length则补0
# padding='pre'在句子前面进行填充，padding='post'在句子后面进行填充
X = pad_sequences(sequences, maxlen=max_length, padding='pre')
```

使用先前创建的 Tokenizer 对象对文本进行编码，将每个文本转变为一个数字序列。该代码首先调用 tokenizer 对象的 texts_to_sequences 方法，将文本中的每一个单词都对应到该单词在 tokenizer 库词典中的整数编号。同时，该代码使用了 max_length 变量来对每个句子进行填充或者截断。如果某个句子的长度大于 maxlen，则该句子将被截断，仅取前 maxlen 个单词。相反地，如果某个句子的长度小于 maxlen，则在句子前面进行 padding（填充），用 0 来补齐序列长度。需要注意的是，填充的位置可以在句子前面'pre'或者在句子后面'post'进行，这可以通过 padding 参数来设置。

最终的输出是经过编码和填充后的文本数据集 X，X 中的每一个元素都是一个定长的数字序列，它们是对原来的文本进行编号和填充后得到的结果。注意，如果某一个单词不在 Tokenizer 库词典之中，则会被过滤掉。如果某个文本被分成的单词数目大于 max_length，则多余的单词将被截断，反之则在句子前添加适当数量的 0 来达到指定的长度。

```
#########step3-定义标签切分数据#########
# 定义标签
# 01为正样本，10为负样本
positive_labels = [[0, 1] for _ in range(poslen)]
negative_labels = [[1, 0] for _ in range(neglen)]
# 合并标签
Y = np.array(positive_labels + negative_labels)
# 切分数据集
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)
```

```
#########step4-搭建模型#########
# 定义函数式模型
# 定义模型输入，shape-(batch, 202)
sequence_input = Input(shape=(max_length,))
# Embedding层，30000表示30000个词，每个词对应的向量为128维
embedding_layer = Embedding(input_dim=30000, output_dim=embedding_dims)
# embedded_sequences的shape-(batch, 202, 128)
embedded_sequences = embedding_layer(sequence_input)
# 双向LSTM
x = Bidirectional(LSTM(lstm_cell))(embedded_sequences)

# 全连接层
x = Dense(128, activation='relu')(x)
# Dropout层
x = Dropout(0.5)(x)
# 输出层
preds = Dense(2, activation='softmax')(x)
# 定义模型
model = Model(sequence_input, preds)
```

建模主要包括以下几个步骤：

1.切分数据集：将整个数据集切分为训练集和测试集，其中训练集包含 80%的样本，测试集包含 20%的样本。

2.定义模型输入：定义了一个形状为(batch,202)的模型输入层，也就是模型的输入数据是一个 batch 中每个样本由 202 个词组成的文本。

3.进行词嵌入：通过 Embedding 层将每个词转换成 128 维的词向量，其中 input_dim=30000 表示总共有 30000 个词，output_dim=128 表示每个词转换成的词向量是 128 维的。

4.双向 LSTM：使用双向 LSTM 对词向量进行处理，其中 lstm_cell 表示 LSTM 的隐藏单元数。

5.全连接层：对 LSTM 的输出进行全连接层变换，将输出的维度变为 128 维并使用 ReLU 作为激活函数。

6.Dropout 层：为了避免过度拟合，使用 Dropout 随机地丢弃一部分神经元。

7.输出层：使用 softmax 作为激活函数，将该层的输出转换为每个类别的置信度。

8.定义模型：通过 Model 函数定义整个模型，参数依次为输入层和输出层。

```
Train on 955 samples, validate on 239 samples
Epoch 1/3
955/955 [==============================] - 162s 170ms/sample - loss: 0.0987 - acc: 0.9885 - val_loss: 0.1376 - val_acc: 0.9791
Epoch 2/3
955/955 [==============================] - 154s 161ms/sample - loss: 0.0536 - acc: 0.9927 - val_loss: 0.1141 - val_acc: 0.9791
Epoch 3/3
955/955 [==============================] - 153s 161ms/sample - loss: 0.0397 - acc: 0.9916 - val_loss: 0.1672 - val_acc: 0.9791
```

数据集被拆分成了训练集和验证集，分别有 955 和 239 个样本。在训练过程中，模型分别进行了 3 轮迭代(epoch)。模型在训练集上的平均损失(loss)为 0.0987，准确率(acc)为 0.9885，在验证集上的平均损失为 0.1376，准确率为 0.9791。同样的方式，模型在第二轮和第三轮迭代期间的表现也进行了记录。其中，训练集和验证集的数据在每轮迭代周期中也会被重新随机划分，以避免过拟合。

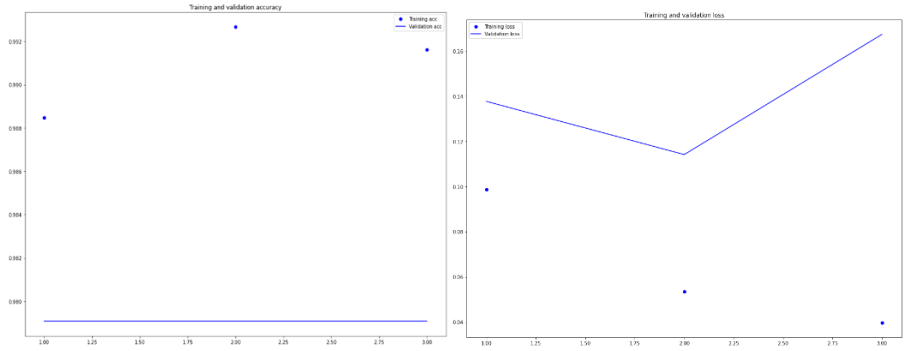


图 3-9 模型的训练准确率和损失函数

在这个 lstm 模型中，训练了三个 epoch，每个 epoch 表示用所有的训练数据更新了一次模型参数。损失函数和准确率是我们对训练模型效果的评估指标，损失函数表示模型的误差大小，准确率表示模型对训练数据的分类或者回归的准确程度。在这个模型中，可以看到训练集和验证集的损失函数和准确率都比较稳定，在三个 epoch 之后准确率仍然保持在 97.9%左右，说明这个模型在处理这些数据方面具有很好的效果。同时，可以看到验证集的损失函数在第三个 epoch 增加了一些，这可能表示模型有一些过拟合，即在训练集上拟合得太好，在验证集上效果有所下降。

结语

随着国内电影市场的蓬勃发展，国产电影也逐步实现了多样性和个性化的发展，并激发了评价和观众反馈的研究热点。在这个背景下，文本挖掘技术为我们提供了一个强大的工具，能够通过计算机技术对大量的文本数据进行分析和信息提取。在电影市场中，通过对电影评论、社交媒体等文本数据的分析，我们能够挖掘出影片的特点、受众对电影的反响以及口碑等信息。这对于电影制作方来说，能够帮助他们更好地掌握市场趋势和观众需求，从而提高电影的口碑和票房。而对于电影评论者和观众来说，文本挖掘技术也能够提供更准确的反馈和评价，以

达到更好的文化传播和理解效果。