

学号：201910210016

# 上海海事大学

## 本科毕业论文（设计）



论文题目： 基于爬虫的评论文本分析系统

姓 名： 王 涵

学 院： 信息工程学院

专 业： 网络工程


班 级： 网络 192 班

指导教师： 张 琳

完成时间： 2023 年 5 月

# 承 诺 书

本人郑重承诺：所呈交的《基于爬虫的评论文本分析系统》是在导师的指导下，严格按照学校和学院的有关规定由本人独立完成。文中所引用的观点和参考资料均已标注并加以注释。论文研究过程中不存在抄袭他人研究成果和伪造相关数据等行为。如若出现任何侵犯他人知识产权等问题，本人愿意承担相关法律责任。

承诺人（签名）：

时 间： 2023 年 5 月 29 日

## 摘 要

随着互联网技术的不断发展，文本分析技术也得到了越来越广泛的应用。而对于电影爱好者来说，提前了解一部电影的评价和口碑显得尤为重要。因此，本文提出了一个基于爬虫的评论文本分析系统的设计和开发，以满足人们对于通过大数据了解电影质量的需求。该系统可以自动抓取豆瓣网站上的电影评论数据，并进行自然语言处理、情感分析等操作，最终输出情感分析结果。这不仅能够帮助用户更好地选择观看的电影，同时也为电影制作方提供了重要的参考依据。

本次设计开发的基于爬虫的评论文本分析系统，综合运用 Python 编程语言、爬虫数据爬取、文本分析和数据库开发等技术，系统主要包括 Python 爬虫模块，文本分析模块，可视化模块，具有数据爬取、文本分析、数据可视化等功能。其中文本分析采用了机器学习算法中的朴素贝叶斯算法，通过对该算法的训练和测试，实验结果表明该算法有较好的稳定性和可靠性，使得本系统有较好的可靠性和可行性。

该系统具有界面美观、功能齐全、交互体验好等优点，实现了爬虫爬取文本数据的功能、文本情感分析的功能以及数据可视化的功能。该系统能为用户提供方便快捷的分析评论文本情感倾向的功能，让用户能更便捷地了解一部电影的评价。

**关键词：**Python 爬虫；文本分析；数据可视化

## Abstract

With the continuous development of Internet technology, text analysis technology has been widely used. For movie lovers, it is particularly important to know the evaluation and reputation of a movie in advance. Therefore, this thesis proposes the design and development of a crawler-based comment text analysis system to meet people's needs for understanding movie quality through big data. The system can automatically capture movie review data from Douban website, and perform natural language processing, sentiment analysis, and other operations to output sentiment analysis results. This can not only help users choose movies to watch better, but also provide important reference for movie makers.

The crawler-based comment text analysis system developed this time comprehensively uses technologies such as Python programming language, crawler data acquisition, text analysis, and database development. The system mainly includes Python crawler module, text analysis module, and visualization module, with functions such as data acquisition, text analysis, and data visualization. The text analysis uses Naive Bayes algorithm in machine learning algorithms. Through the training and testing of this algorithm, experimental results show that the algorithm has good stability and reliability, making this system practical and reliable.

This system has the advantages of beautiful interface, complete functions, and good interactive experience. It realizes the functions of crawling text data acquisition, text sentiment analysis, and data visualization. The system can provide users with a convenient and fast way to analyze the sentiment tendency of comment texts, allowing users to easily understand the evaluation of a movie.

**Key words:** Python crawler; Text analysis; Data visualization

# 目 录

摘 要 .....	I
Abstract.....	II
1 概述 .....	1
1.1 研究背景及意义 .....	1
1.2 国内外研究现状 .....	1
1.2.1 国内研究现状 .....	1
1.2.2 国外研究现状 .....	3
1.3 主要工作及论文组织结构 .....	4
1.3.1 研究内容 .....	4
1.3.2 研究思路 .....	4
1.3.3 论文组织结构 .....	5
2 系统开发环境及相关技术 .....	6
2.1 系统运行硬件环境 .....	6
2.2 系统运行软件环境 .....	6
2.3 相关技术 .....	6
2.3.1 Python 爬虫技术 .....	6
2.3.2 文本处理 .....	7
2.3.3 文本分类技术 .....	8
2.3.4 可视化技术 .....	9
2.3.5 SQLite 数据库 .....	10
3 系统设计 .....	11
3.1 需求分析 .....	11
3.2 系统总体设计 .....	11
3.3 数据库设计 .....	12
3.3.1 数据库表设计 .....	12
3.3.2 数据库表结构 .....	13
4 系统实现 .....	16
4.1 Python 爬虫 .....	16
4.2 文本预处理 .....	20
4.2.1 加载语料 .....	20
4.2.2 分词和清洗 .....	21
4.3 基于朴素贝叶斯的情感分析 .....	21
4.4 可视化 .....	24
4.4.1 Echarts .....	24

4.4.2 词云图 .....	25
5 系统测试及结果分析 .....	27
5.1 实验分析与结果 .....	27
5.2 系统测试 .....	29
6 总结与展望 .....	38
6.1 总结 .....	38
6.2 展望 .....	38
参考文献 .....	40
致谢 .....	42

# 1 概述

## 1.1 研究背景及意义

近年来,随着中国国民经济水平的不断提高,以及我国已经全面进入小康社会阶段,越来越多的人开始重视精神上的满足。读一本好书,看一部好电影,让人的心灵得到洗涤,情感得到丰富,并带来对人生的思考。而看一部电影,既能够满足自己的精神需求,也能够和亲朋好友一起分享快乐。因此,越来越多的人选择看电影,通常会根据互联网评论选择可看的电影,并且愿意在互联网中给自己看过的电影给出评价。

随着信息时代的到来,以及在互联网的高速发展下,越来越多的人能够熟练的使用互联网发布和搜索数据。根据中国互联网络信息中心(CNNIC)发布的第 50 次《中国互联网络发展状况统计报告》显示,截止 2022 年 6 月,我国的网民规模达到 10.51 亿,互联网普及率达到 74.4%。正是由于越来越多的人使用互联网,导致互联网中充满了大量的、各种各样的数据,因此也推动了大数据的发展。同时也产生了大量值得人们去获取和分析的数据。

在互联网的大量数据中,产生的文本数据是值得人们研究的一类数据,因为某些文本数据可能会带来经济价值或社会价值,可能会为企业和机构提供更好的数据分析服务和决策支持。比如,通过对微博等社交媒体的评论文本分析就能知道一些当下的社会舆论焦点。在新闻报道中,通过对全网新闻进行数据抓取和分析,可以了解各种热门话题的热度、舆情数据和事件的发展趋势。通过对购物网址的商品评论文本分析就能够知道这件商品的大众满意度,以达到为企业提供更精准的市场营销方案的目的。而通过对电影评价的分析,既能够为我们带来一些对于此电影的参考价值,也能够反映出一些当下的关注热点。基于这些现状,我们可以察觉到文本数据分析在当今数据分析领域中具有重要的地位和应用前景,它的研究和发展也会不断推动着技术的进步和社会的发展。每当有一部新电影上映,就会产生大量的影评文本数据,这些数据往往是爆炸式增长的,单靠人工去收集整理,成本过高。因此,本文希望以此为切入点,通过网络爬虫的技术手段对豆瓣平台上的电影影评文本数据进行分析。

## 1.2 国内外研究现状

### 1.2.1 国内研究现状

网络爬虫,又称网页蜘蛛,是一个功能强大的能够自动提取网页信息的程序,它模仿浏览器访问网络资源,从而获取用户需要的信息,它可以为搜索引擎从万维网上下载网页信息,因此也是搜索引擎的重要组成部分<sup>[1]</sup>。正如人们所认知的那样,爬虫最早的用途是服务于搜索引擎的数据收集,而现代意义上的搜索引擎的鼻祖是 1990 年由加拿大麦吉尔大学学生 Alan Emtage 发明的 Archie。后来越来越多的搜索引擎也相继出现,如百度、谷歌等。如今,爬虫技术已经广泛用于大数据行业,也成为了一项热门的话题。

杜晓旭等人通过使用爬虫技术对微博中的数据进行分析，通过使用关键词匹配模块，做到了使匹配更精准，进而能使其他开发者对微博数据进行更深入的分析，并提出了爬虫技术具有较高的实用性和有效性<sup>[2]</sup>。

文本分析就是对文本数据进行分析，从非结构化的文本中对其特征进行挖掘以及对其特征进行统计分析。文本分析又可以分为情感分析、主题分类、问答任务、意图识别、自然语言推理等不同的种类和方向。其中，文本情感分析，是指使用文本分析等数据分析技术对文本数据所包含的观点和情绪等内容进行分析和判断，以实现文本的情感极性做出分类和判断。随着计算机技术的发展，人们逐渐地研究和发展出了如 NLP、GPT 等自然语言处理模型从而让计算机理解对自然语言的处理。文本情感分析作为自然语言处理领域的重要研究组成部分，它包含了语言学、心理学、统计学和人工智能等不同领域的理论与研究方法<sup>[3]</sup>。因此，对于文本情感分析来说，首先需要对爬取到的自然语言文本数据进行加工处理，然后对它处理后的数据进行分类。我们可以将文本数据分成客观性文本和主观性文本两类。前者包括人类语言对物体或事件的客观性描述；后者则是人类语言对物体或事件主观评价，通常包含主观意见、情感、观点和态度等。将主观语言的文本抽取出来，过滤掉不带情感色彩的文本，为文本情感分析提供主观性文本数据。在完成上述步骤后，就需要分析过滤后的主观性文本，这里又主要包括文本情感极性分析和文本情感极性强度分析。前者是去识别和判断主观文本的情感极性（通常是正面的赞赏和肯定、负面的批评与否定，但也有一些学者考虑文本极性并非是非黑即白，因此加入了中性这一极性判断）。

对于文本情感分析的研究方法主要有三种，分别是基于情感词典的方法、基于机器学习算法的方法和基于深度学习算法的方法。在使用词典法方面，郑诚等人通过构建情感词典，并将其与语义规则组合起来，结合否定词表与程度副词表，利用加权求和的方式给出每个情感词组合的分值，并建立微博情感分析模型<sup>[4]</sup>。高华玲等人使用了情感词典的情感文本分析方法，对用户评论进行情感倾向性分析，提出了使用基于领域情感词典的方法比通用词典在专业领域内更具优势<sup>[5]</sup>。刘博等人通过使用英文情感词典，设计了基于语义关系的情感词典自动构建方法，并且借助英文情感词典的构建方法对中文情感词典进行了构建，并对比和验证了现有的常用情感词典<sup>[6]</sup>。在使用机器学习算法方面，刘志明等人通过分别采用三种机器学习算法、特征选取算法以及特征项权重计算方法对微博评论文本进行了情感分类的实证研究<sup>[7]</sup>。李春林等人对爬取到的白酒吧股评文本数据先进行数据预处理、中文分词、情感分析、绘制词云图等操作，通过对比六种机器学习算法进行情感分类模型拟合的准确程度，得出 SVM、KNN 对情感分类的准确度更高的结论<sup>[8]</sup>。葛霓琳等人采用了朴素贝叶斯及 SVM 两种基于机器学习的分类方法对文本数据进行了情感分析<sup>[9]</sup>。李艳红提出基于信息字节 N 元语法、信息量、评论文本的情感极性等对特征进行扩展的方法，并结合 SVM 算法，对自然语言文本进行了观点挖掘<sup>[10]</sup>。向志华基于机器学习的几种文本分类技术进行了系统的研究，并指出了文本分类技术的未来发展方向<sup>[11]</sup>。在使用深度学习算法方面，黄贤英等人通过使用双向 LSTM 神经网络



络获取更为完整的文本上下文信息从而提取出深度词向量特征，继而使用基于 one-versus-one 的支持向量机对其进行情感分类<sup>[12]</sup>。王汝娇等人对 Twitter 文本词向量使用卷积神经网络获得对应的深度词向量特征，将各类特征进行特征融合并采 One-Versus-One SVM 实现情感极性的分类判别<sup>[13]</sup>。梁军等人使用递归神经网络来发现与任务相关的特征，并根据每段文本数据的词语前后的相关性引入情感极性转移模型，从而加强了对文本关联性的捕获<sup>[14]</sup>。刘艳梅使用网络爬虫技术从微博上抓取部分数据，经过词料预处理后输入卷积神经网络，并基于 SVM/RNN 等机器学习算法构建了分类器，并在测试集中判断了每句自然语言文本的情感倾向<sup>[15]</sup>。马文等人通过进行了一系列的实验对比得出朴素贝叶斯分类器在采用集成特征选取时文本分类的准确率最佳，验证了朴素贝叶斯分类器在处理中文评论分类问题的实用性与可行性<sup>[16]</sup>。

### 1.2.2 国外研究现状

在国外，类似爬虫等技术都是来源于国外，因此国外对文本分析的研究也有很多。大家也都基本上使用上述的三种方法进行研究。Ramasamy 等人提出了一种新颖的基于信息增益的特征选择算法，通过删除不合适的内容来选择高度相关的特征。利用该算法，在文档级、句子级和特征级进行了广泛的情感分析<sup>[17]</sup>。Shailendra 等人采用了一种包含意见动词及其情感分数意见的动词词典，对社交媒体的文本中存在的意见动词的文本情感进行了分析<sup>[18]</sup>。Christopher 等人则构建了 WKWSCI 情感词典，在产品评论文本的情感分类、新闻头条内容的情感分类任务上都能有出色的表现和良好的性能<sup>[19]</sup>。Qiu 和 Zheng 通过分析了 2010 年 10 月 1 日至 2013 年 6 月 30 日期间北美电影市场的数据，以研究使用预测组合方法来预测票房表现的潜在好处，并验证了基于正则化的预测组合相比于单个预测者和备选组合方法，能显著提高了预测效果<sup>[20]</sup>。Zhou 等人提出了一种简单的基于一致性正则化的多模型对比学习算法 Multi-MCCR，可以有效缓解模型训练和推理之间由于随机信息缺失带来的不一致性，又将该模型运用于情感分析、主题分类和评论分类并验证了模型的有效性<sup>[21]</sup>。Zhao 等人提出了一种框架以改进现有的情感分析中的 DNN 模型<sup>[22]</sup>。Waheeb 提出了一种新的基于向量空间模型、统计方法、关联规则和极限学习机自编码器的情感分析方法并将测试集通过该方法和其他现有方法进行对比，验证了其提出的方法是有效的<sup>[23]</sup>。Bagla 等人提出了一种信息文本分析模型 TA-WHI，使用 CNN、LSTM 和 CatBoost 等深度学习分类器评估模型的性能。所设计的模型在 CatBoost 中达到了 98 % 的准确率<sup>[24]</sup>。

通过对国内外技术现状的了解，在如今这个大数据的背景下，互联网中还有很多有价值的数据等着我们去发现，等着我们去挖掘，让其产生更大的价值。通过对某一部电影的影评文本进行分析，能够更清楚直观的了解电影的相关情感信息，也能给电影界对于这部电影做出更多的参考价值，同时也可以帮助电影公司和电商企业洞察电影市场的趋势和规律，了解用户对于不同类型电影的评价和反馈，从而制定更合适的宣传和营销策略。通过将文本数据经过分析之后，以可视化的方式展现给大众，让人一目了然的知

道电影的相关信息，以及网友对该电影的评价是积极、中性还是消极。

### 1.3 主要工作及论文组织结构

#### 1.3.1 研究内容

综合国内外文本分析的现状，本文针对目前互联网上充斥的大量文本数据，通过爬虫技术对其进行爬取并针对爬取到的文本进行分析。文本拟以豆瓣网站上的电影相关影评文本数据作为研究对象进行文本分析研究。研究主体内容将包括以下几个部分：

（1）Python 网络爬虫：通过对 Python 语言和网络爬虫的学习，达到能够将所需的对象成功爬取并保存下来，以便于后续的分析处理。

（2）文本分析：将爬取下来的文本数据进行分类处理，使每一类数据都在数据库中清楚地展示出来。再将影评文本数据进行拆分、归类，通过情感文本分析的各种方法将每一条评论文本都能够划分到正确的类别里面。

（3）数据可视化分析：在文本数据分析完成之后，将数据结果通过可视化的方式直观地展现出来。

（4）数据库技术：用于存储爬取下来的文本数据以及其他的一些相关信息。

#### 1.3.2 研究思路

对目前已有的一些在对 Python 爬虫、文本分析、情感分析、自然语言处理等方面的文献进行阅读学习和总结，确定论文的研究对象和整体的研究思路。再通过学习爬虫技术，学习如何将所需要的文本获取下来，并将其保存在数据库中。再对获取和保存的文本数据进行自然语言分析处理，包括文本情感分析，最后再将其可视化展现出来，通过 Web 页面访问查看。

本文所设计的系统主要包括文本数据获取与存储、文本处理、文本分析、数据可视化和 Web 页面展示五个部分。如图 1-1 所示。第一部分，文本数据获取与存储，即通过爬虫技术在豆瓣网站上爬取所需要的文本数据，并将其保存到 SQLite 数据库中。第二部分，文本处理，即通过相关技术实现文本的预处理，主要包括语料加载、加载自定义分词、分词、数据清洗等操作。加载语料，即将爬取到的数据从数据库中加载到程序中。加载自定义分词，即加载用户自定义的分词，便于分词操作。分词，即将文本数据分为一个一个词语。数据清洗，即将分词后的文本删去标点符号和停用词等操作。第三部分，文本分析，即通过机器学习的方法训练模型实现文本分析。本系统主要采用朴素贝叶斯算法来实现，通过计算特征值矩阵以及训练朴素贝叶斯分类器以实现文本情感分析的目的。第四部分，数据可视化，即通过词云图、Echarts 饼图等实现可视化效果以达到直观简洁的可视目的。第五部分，Web 页面展示，即通过网页展示系统。其中使用 Python 作为后端编程语言，前端编写 HTML 页面，使用 Flask 框架实现前后端数据交换和功能实现。

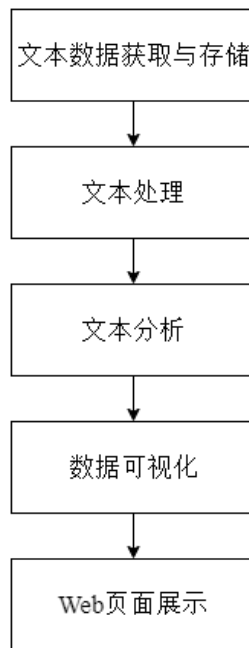


图 1-1 系统架构图

### 1.3.3 论文组织结构

第一章，绪论。主要简单介绍研究背景与意义，国内外研究现状，论文的主要研究内容，系统架构，论文研究方法，以及论文的组织结构。

第二章，系统开发环境及相关技术。主要介绍系统的运行软硬件环境，以及本文所使用的一些相关技术。

第三章，系统设计。主要介绍本文所设计系统的需求分析和系统总体设计，以及数据库的相关设计。

第四章，系统实现。主要介绍了系统所涉及的主要四个模块，详细介绍了整个系统的关键技术。

第五章，系统测试及结果分析。主要将实验分析与结果和系统运行测试进行展示。

第六章，总结与展望。总结全文主要内容，并提出不足与展望。

## 2 系统开发环境及相关技术

### 2.1 系统运行硬件环境

处理器：Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz    2.30 GHz 基于 x64 的处理器

内存：16GB DDR4 2666MHz

硬盘：512GB SSD 固态硬盘

### 2.2 系统运行软件环境

操作系统：Windows 11 家庭中文版

开发环境：PyCharm Community Edition 2022.2.4

开发语言：Python 3.11.0

数据库：SQLite 3

### 2.3 相关技术

#### 2.3.1 Python 爬虫技术

Python 爬虫技术通过模拟人的行为访问互联网，通过访问网址获取当前网址的源代码，即向网站的服务器发送一个请求，返回的响应体便是网页源代码。将网页源代码解析出来，再分析源代码获取到我们需要的数据。其中，Python 提供了许多库文件帮助我们便捷地使用爬虫技术。使用 Urllib、Requests 等库实现 HTTP 响应获取网址源码；使用 BeautifulSoup 等库，让我们可以高效快速地从中提取网页信息，如节点的属性、文本值等；使用 re 库通过正则表达式提取所需的指定内容。其中最主要的就是 Requests 和 BeautifulSoup 库。Requests 是一个用于发起网络请求的 Python 第三方库，在使用时，可以方便地发起 HTTP 请求，如 GET、POST、DELETE、PUT 等，并获取响应结果。BeautifulSoup 是一个用于解析 HTML 和 XML 文件的 Python 第三方库。它可以用来处理正则表达式或者 XPath 语言难以直接读懂的页面结构，并提取所需的信息。

简单的爬虫架构由五部分组成，Python 爬虫架构图如图 2-1 所示：

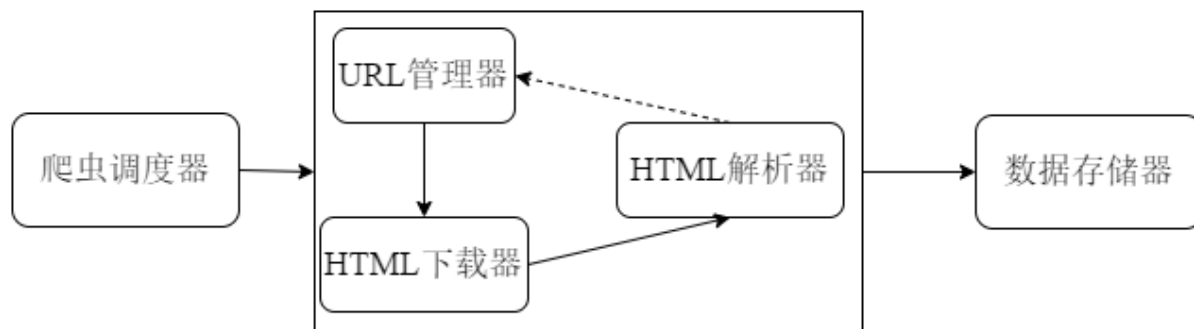


图 2-1 爬虫架构图

分别是爬虫调度器、URL 管理器、HTML 下载器、HTML 解析器、数据存储器。其中爬虫调度器总体协调其它几个模块的工作。URL 管理器负责管理 URL，维护已经爬取的 URL 集合和未爬取的 URL 集合。网页下载器对未爬取的 URL 下载。网页解析器解析已下载的 html，并从中提取新的 URL 交给 URL 管理器，数据交给存储器处理。数据存储器将 html 解析出来的数据进行存取。

一个典型的 Python 爬虫应用往往包含以下几个步骤：首先获取目标 URL，使用 Python 的 HTTP 库或第三方库（如 Requests）来获取目标 URL 响应的 HTML 代码。然后解析 HTML 代码：使用 Python 的解析库（如 BeautifulSoup、lxml）来解析 HTML 代码，从中筛选出需要的数据。再存储数据：使用 Python 的数据库（如 MySQL、pymongo）将数据存储到数据库中，或使用 Python 的文件库（如 csv、json）将数据存储到文件中。最后进行可视化处理：使用 Python 的数据可视化库（如 matplotlib、Seaborn）对爬取到的数据进行可视化处理。如图 2-2 所示：

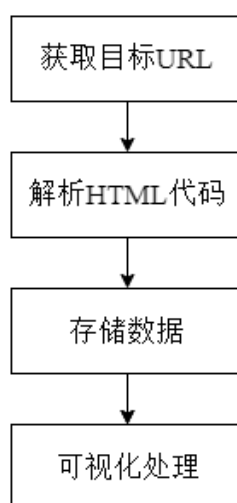


图 2-2 爬虫执行流程图

### 2.3.2 文本处理

文本处理是指对文本数据进行处理、分析、挖掘，提取有价值的信息和知识。它是一种数据挖掘分析的方法，关注文本中隐藏的关联、规律和趋势。它主要是指对文本的表示及其特征项的选取。文本处理是文本挖掘的基础，通过把文本中抽取出的特征词进行量化来表示文本信息。根据文本数据分析的内容，可以推断出文本所要表达的意图和目的。其步骤主要包括：文本预处理、数据清洗、特征提取。

#### （1）文本预处理

文本预处理是一系列针对原始文本数据的操作，目的是为了将其转换为可用于文本分析和机器学习模型的规范化、标准化的格式。这些操作通常能为文本分类、情感分析、实体识别和自然语言处理的任务提供更好的性能。它主要包括分词操作，即将原始文本分割成单词或短语，以便进行下一步处理。

## （2）数据清洗

数据清洗是指对数据进行过滤、清理、转换和修复等一系列处理，以去除或修正数据中存在的错误、不规则和不完整等问题。数据清洗通常是紧接着数据预处理的一步，也是确保数据质量和可靠性的重要步骤。它主要包括以下操作：

去除停用词，将常见的无意义单词和词语（如“的”、“和”、“而且”等）从文本中去除，以减少数据噪声。去除标点符号，将文本中的标点符号（如“，”，“。”，等）去除，以减少噪声和规范化数据。清理异常值，检测并处理异常值数据，以避免影响模型性能和结果的准确性。

## （3）特征提取

特征提取是指从原始数据中抽取出有用的信息，以作为机器学习模型的输入。在机器学习中，特征提取是非常重要的步骤，因为它直接关系到模型的表现和准确性。在文本处理中，它主要用于提取关键字等操作。在实际运用中多使用 TF-IDF 实现特征提取。TF-IDF（词频-逆向文档频率）可以用来评估单词对于文档的重要程度。

### 2.3.3 文本分类技术

本文所涉及的文本分类技术也指文本分析、文本情感分析、情感分类。常见的文本分类算法主要包括机器学习算法和深度学习算法。其中机器学习算法主要包括决策树算法、支持向量机（SVM）算法、K 最近邻（KNN）算法、朴素贝叶斯算法等。深度学习算法主要包括神经网络算法等。

#### （1）决策树算法

决策树算法就是通过对各个特征进行划分，建立一棵树型结构的模型，从而实现对数据的分类，是一种有监督学习的算法。其主要是树形结构通过 if-then 规则判断，采用“自上而下分而治之”的思想策略。决策树算法易于理解，生成的树形结构清晰直观，并且不需要归一化。但是它容易产生过拟合问题，特别是最大深度不受限时容易出现过拟合。过拟合就是指在训练时结果表现优秀，而在测试时表现较差。同时由于决策树模型较为简单，通常不能达到最高精确度，无法很好地适用于某些复杂问题。以及当特征之间的关联性很强时，决策树算法容易出现欠拟合问题。欠拟合是指在训练集和测试集上表现都较差。

#### （2）支持向量机（SVM）算法

支持向量机的基本思想是通过构建最优分类超平面来实现对数据的分类。该算法的关键是选择合适的超平面，在超平面的左右两边分别代表不同的类别。它对于高维数据集和非线性数据集有着较好的表现，在获取更多样本数据时对算法不会有太大的潜在影响，通常也不会出现过拟合现象。但它也存在一些缺陷，如模型复杂度高、内存消耗大，计算量大等，处理噪声数据等不良数据表现不如其他机器学习算法。

#### （3）K 最近邻（KNN）算法

KNN 算法就是对于一个未知类别的样本，通过找出训练集中最近 k 个样本的类别

进行投票，得票最多的类别即为该未知样本所属的类别，即通过邻近的几个样本特征来判断。因此所有特征都需要作可以比较的量化。KNN 算法思想比较简单，易于理解和实现，在处理多分类问题方面表现更好。但是，KNN 算法计算时需要较大的内存空间，计算成本高，效率低，并且准确率受特征值选取影响较大。当某一类的特征值较多时，计算结果会更偏向于该分类，因此，噪声或异常值的存在会对算法的分类结果产生较大影响。

#### （4）朴素贝叶斯算法

朴素贝叶斯算法是一种基于贝叶斯定理和特征条件独立假设的分类算法，它来源于严密的数学模型，并基于构建特征的条件概率分布，模型简单，易于实现和理解。当有新的文本数据加入时，朴素贝叶斯算法可以通过增量学习方法，快速适应新的数据，且不会影响整体模型。因此，可以通过增加训练数据提高准确度。并且由于条件独立性假设这一特性，其所需计算的参数较少，计算效率较高，能够快速计算出结果。最主要的是在小数据样本中，它的表现优于其他算法。而且对于缺失数据和异常数据的分类问题，朴素贝叶斯依然可以保持较高的准确率。

#### （5）神经网络算法

神经网络算法就是通过一定的网络结构和训练方法来模拟生物神经元间的相互作用。主要包括卷积神经网络(CNN)、循环神经网络(RNN)、长短期记忆神经网络(LSTM)等。它可以通过深度学习自动学习特征，不需要手动设计特征，并行能力强，适合处理大规模数据。但同时它也需要大量的计算资源，需要大量的数据进行计算，如果数据量过少，训练结果的可行度不高。也如 KNN 算法一样，噪声或异常值的存在会对算法的分类结果产生较大影响，因此也容易出现过拟合或欠拟合的情况。

由于豆瓣平台的局限性，本系统通过爬虫爬取的数据只有大约 28000 条评论文本，并且朴素贝叶斯算法简单、高效、快速，适用于本系统的文本分析并快速给出分析结果的模块，因此本系统采用朴素贝叶斯算法来进行文本分析。

### 2.3.4 可视化技术

采用 WordCloud 可视化技术，WordCloud 可以用于生成词云。词云以词语为基本单位，根据词语的出现频率确定词语的大小，将所有这些词放到一张图片里，就可以更直观和艺术的展示文本。WordCloud 通常使用在文本挖掘和数据分析领域。它通过先对文本数据进行统计和处理，然后将每个词汇按照频率大小转化成大小不等的矩形，再将这些矩形按照一定规则排列在一起，形成一个由文字组成的云状图形。

采用 Echarts 可视化技术，Echarts 是一个基于 JavaScript 的开源可视化库，支持各种常见的图表和图形的绘制，如折线图、柱状图、饼图、散点图、地图、箱线图等。Echarts 提供了丰富的图表效果和交互功能，可以为数据可视化提供强大的支持。本系统采用 Echarts 绘制饼图以达到可视化效果。可以直观了解到评论分布情感以及评论占比情况，让用户一目了然。

### 2.3.5 SQLite 数据库

本系统采用 SQLite 数据库，SQLite 是一个进程内的库，实现了自给自足的、无服务器的、零配置的、事务性的 SQL 数据库引擎。它是一个零配置的数据库，这意味着它不需要在系统中配置。SQLite 是一种轻型的、基于文件的、跨平台的嵌入式关系型数据库系统，它的设计目标是嵌入式设备、移动设备和小型应用程序的数据库系统。SQLite 是使用 C 语言编写的，因此它使用起来非常快速、可靠、高效。它支持 SQL 标准，包括事务、索引、触发器等功能。它还支持许多编程语言，如 C、C++、Java、Python 等。SQLite 的流行程度在移动开发领域很高，它被广泛地应用于 Android 和 iOS 应用程序中。所以，SQLite 是一种轻量级、灵活、易于使用的数据库引擎，它适用于大多数应用程序的需求。



## 3 系统设计

### 3.1 需求分析

本文所设计的系统能够为用户提供电影的相关信息和电影评论的情感极性分析，以及提供帮助用户判断情感极性文本的功能。同时还满足用户数据分析功能需求、系统易用性需求、灵活性需求。系统易用性方面，用户需要一个友好的用户界面，能够方便地进行电影评论的查询和分析，并且界面美观、简洁，易于使用。数据分析功能方面，将豆瓣网抓取到的电影评论进行分析，便于研究和分析关于电影内容和市场的各种情况和关键词等。灵活性方面，用户需要系统能够满足自己的个性化需求，例如，用户可以自己搜索相关电影信息，以便更准确、全面地获取所需信息。

综上所述，为了满足基于爬虫的豆瓣影评文本分析系统的用户需求，系统需要满足易用性、数据分析功能和灵活性等方面的需求。

### 3.2 系统总体设计

本文设计的系统功能主要包括 Python 爬虫、文本分析和数据可视化模块。

爬虫模块主要通过 Python 语言并借助其提供的各种各样的库，将所需要的电影影评内容爬取下来并保存到数据库中以供文本分析时使用。

文本分析模块通过对爬取到的评论文本进行分词、清洗和归类，再将处理好的文本数据通过机器学习的方式进行训练和测试得到训练模型并保存，便可以直接使用模型进行测试和分析，其中机器学习方法主要采用朴素贝叶斯算法。

数据可视化模块通过使用 Flask 前端框架，设计了五个网页界面，分别是首页、新电影页、电影评论页、文本分析页和更多电影页。首页通过 JavaScript 渲染得到美观简洁的网页，让人一目了然地了解到该系统的主题。新电影页展示了最新上映的部分电影，通过爬虫爬取相关信息并展示到页面，并且通过点击每一部电影的名称可以跳转到电影评论页。电影评论页显示了通过爬虫爬取部分评论生成的词云图，评论情感极性情况的饼图，以及好评度。并且随机显示 30 条评论文本，当点击每一条评论时，都能生成对应的文本情感分析结果。然后是文本分析页，通过输入想要测试的评论文本，可以得到文本的情感极性结果。最后是更多电影页，在这个页面用户可以通过自己输入电影名字搜索到自己想要了解的电影，然后后端立即使用爬虫爬取豆瓣网址部分短评，并计算出好评度展示到页面。同时也会展示出情感极性情况饼图，以及电影海报。

系统总体设计功能模块如图 3-1 所示。基于爬虫的评论文本分析系统包括 Python 爬虫、文本分析（情感分析）、数据可视化。Python 爬虫包括保存数据到数据库。文本分析（情感分析）包括朴素贝叶斯。数据可视化包括网页前端的五个页面分别为首页、新电影页、电影评论页、文本分析页、更多电影页，并包括词云图和 Echarts 饼图等。

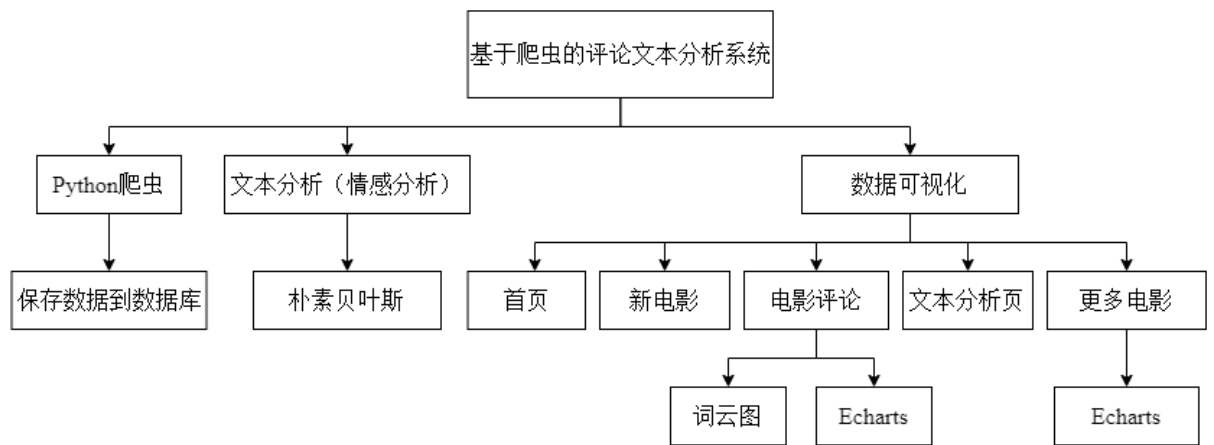


图 3-1 功能模块图

### 3.3 数据库设计

#### 3.3.1 数据库表设计

本文所提及的系统一共使用了六个数据库表，分别为 movie 表、pos 表、neg 表、newmovie 表、moviecomment 表和 Echartsdata 表。其中，movie 表、pos 表和 neg 表是用于机器学习时的表。其 E-R 图如图 3-2 所示。movie 表，即电影表，是用来存放爬虫爬取下来的用于机器学习的评论文本。pos 表，即好评表，是用来单独存放好评的表，它与 movie 表有一一对应的关系。neg 表，即差评表，是用来单独存放差评的表，它同样与 movie 表有一一对应的关系。

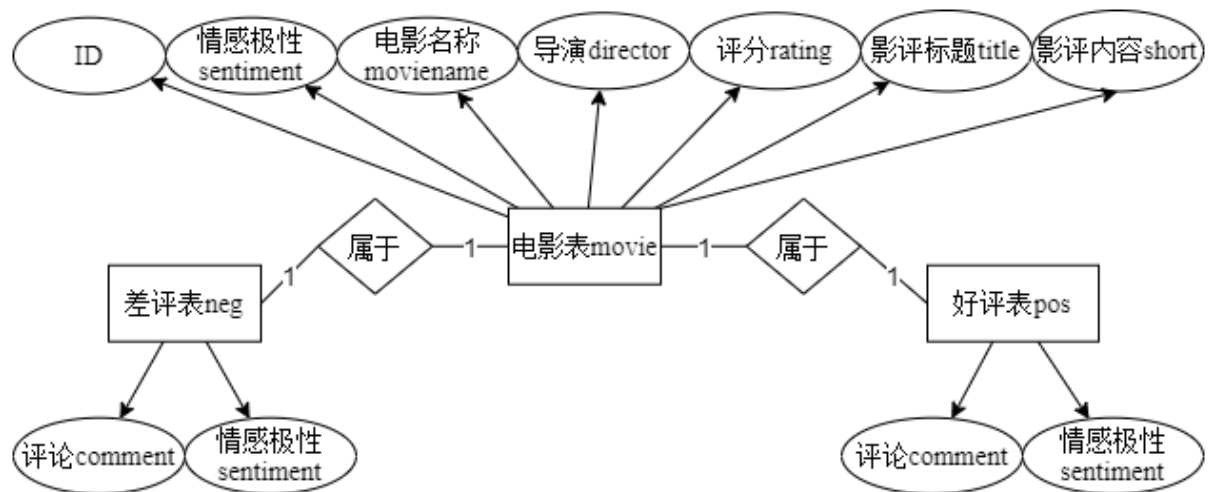


图 3-2 用于机器学习部分的 E-R 图

而 newmovie 表、moviecomment 表和 Echartsdata 表是用于系统前端页面实现的表。其 E-R 图如图 3-3 所示。newmovie 表，即新上映电影信息表，是用来存放最近上映的电影的相关信息。moviecomment 表，即新上映电影评论表，是用来存放新上映电影的部分评论，它与 newmovie 表为一对多的关系，newmovie 表中的每一部电影信息对应多条

评论。Echartsdata 表，即评分表，它用来单独存放电影的评分情况和获取评论的时间。

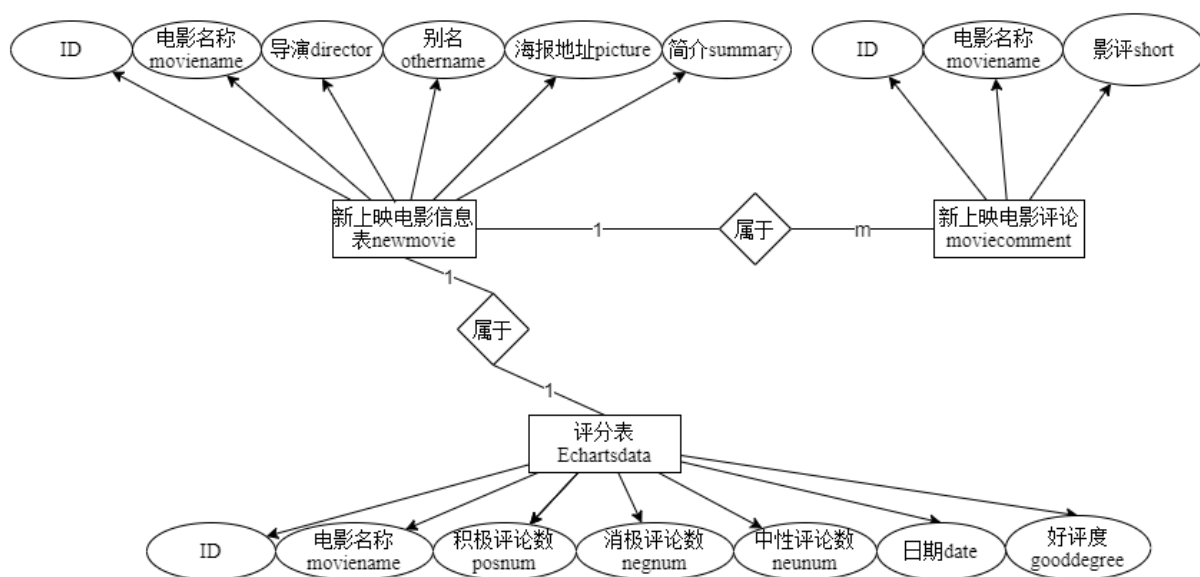


图 3-3 用于系统实现部分的 E-R 图

### 3.3.2 数据库表结构

本文所提及的系统一共有六个主要的数据库表，分别为 movie 表、pos 表、neg 表、newmovie 表、moviecomment 表和 Echartsdata 表。这些表主要用来存储爬虫爬取下来的文本内容以及相关的信息。movie 表结构如表 3-1 所示。movie 表包括七个字段，分别是 id、moviename、director、rating、title、short、sentiment。将 id 设置为主键并让其自增，这样每次在执行插入语句时则会自动生成 id 插入。sentiment 存放情感极性得分，它是根据 rating 分数计算得到的。

表 3-1 movie 表结构

字段名称	类型	说明
id	integer	primary key autoincrement 主键自增
moviename	varchar	电影名称
director	varchar	导演
rating	varchar	评分
title	text	影评标题
short	text	影评内容
sentiment	integer	情感极性

pos 表是单独分离出来的好评表，以便机器学习时使用。它与 movie 表有一一对应的关系，包括两个字段 comment 和 sentiment。pos 表结构如表 3-2 所示。comment 中存放的都是积极评论，sentiment 中则存放其对应的情感极性得分。情感极性得分分为三种情况，-1 分代表消极评论，0 分代表中性评论，1 分代表积极评论。

表 3-2 pos 表结构

字段名称	类型	说明
comment	text	积极评论
sentiment	integer	情感极性

neg 表与 pos 表类似，都是单独分离出来的表，存放差评，以便机器学习时使用。它与 movie 表有一一对应的关系，包括两个字段 comment 和 sentiment。数据库表结构如表 3-3 所示。

表 3-3 neg 表结构

字段名称	类型	说明
comment	text	消极评论
sentiment	integer	情感极性

newmovie 表包括 6 个字段，分别是 id、movienam、director、othername、picture、summary。数据库表结构如表 3-4 所示。将 id 设置为主键并让其自增，这样每次在执行插入语句时则会自动生成 id 插入，movienam 存放的是电影名，director 存放导演名，othername 存放电影别名，picture 中存放的电影海报的地址链接，可以在网页通过网络直接获取到海报图片，并将其展示出来。summary 存放电影相关简介，能够在网页固定位置显示出来。

表 3-4 newmovie 表结构

字段名称	类型	说明
id	integer	primary key autoincrement 主键自增
movienam	varchar	电影名称
director	varchar	导演
othername	varchar	别名
picture	varchar	海报地址
summary	varchar	简介

moviecomment 表是将影评内容单独创建的表，它与 newmovie 表有相关性，一个 newmovie 表中的电影对应 moviecomment 中的多条评论。这里将评论文本单独建立一个表是为了方便查询使用。它包括 3 个字段，分别是 id、movienam 和 short。同样，id 作为主键，设置自增。movienam 存放的是电影名，short 存放影评内容。其结构如表 3-5 所示。

表 3-5 moviecomment 表结构

字段名称	类型	说明
id	integer	primary key autoincrement 主键自增
moviename	varchar	电影名称
short	text	影评内容

Echartsdata 表是用于存储新电影评论的评分信息表，通过对电影评论的分析得到的好评、差评的数量记录在表中，便可计算出好评度。同时记录得到数据的时间，以方便更新网页中的好评度表。它包括七个字段，其中好评度是由 neg 值（差评数量）、pos 值（好评数量）和 neu 值（中评数量）计算得出。数据库结构如表 3-6 所示。

表 3-6 Echartsdata 表结构

字段名称	类型	说明
id	integer	primary key autoincrement 主键自增
moviename	varchar	电影名称
posnum	varchar	积极评论数
negnum	varchar	消极评论数
neunum	varchar	中性评论数
date	varchar	日期
gooddegree	varchar	好评度

## 4 系统实现

### 4.1 Python 爬虫

Python 爬虫是一种伪装成客户端与服务端进行数据交互的程序。它的开发流程一般是通过指定 URL 发送请求并获取响应，在获取到响应后提取相关数据，最后将数据保存下来。爬虫工作流程如图 4-1 所示。

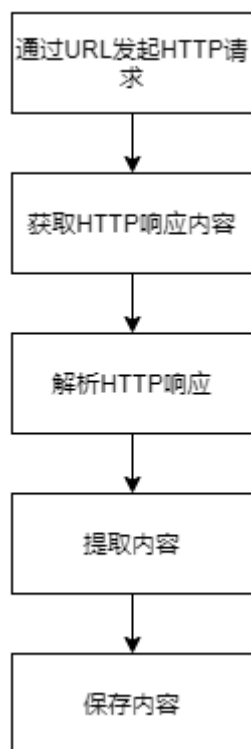


图 4-1 爬虫工作流程图

看似简单的访问、提取、保存这几步，实际在使用时却并不简单，因为许多网址有反爬虫机制。反爬虫就是指通过使用任何技术手段，阻止别人批量获取自己网站信息的一种方式。同时，除了反爬虫机制外还会遇到其他限制，比如访问次数过多便无法访问等等。当然对于这些问题也有许多解决办法，有的方法简单有的复杂。

以爬取豆瓣网的电影相关信息为例，编写爬虫访问函数，首先导入 `urllib` 库，`urllib` 库是 Python 内置的 HTTP 请求库，再使用 `urllib.request.Request()` 方法通过 URL，返回一个 `urllib.request.Request` 对象。此时，在方法中要添加 URL 参数和 `headers` 参数，其中 URL 为必填参数。`headers` 参数为添加请求头，此参数就是为了防止反爬虫机制，如果没有设置 `headers`，那么就会被反爬虫机制拦截，获取资源失败。`headers` 类型为字典，里面需要添加相关的 `User-Agent` 等参数，来防止反爬虫并实现模拟本地浏览器访问网页的过程。然后使用 `request.urlopen()` 方法向网站发起请求并获取响应对象，返回一个 `HTTPResponse` 类型的对象。方法有两个参数分别是 URL 和 `timeout`，其中 URL 是必选

参数，可以是一个字符串或 Request 对象，本文是采用 request 对象。最后通过获取到的响应对象来获取响应内容，并赋值给字符串 html，最后返回 return html。代码流程图如图 4-2 所示。

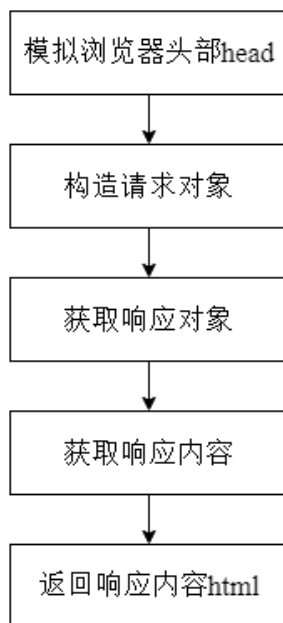


图 4-2 爬虫获取响应代码流程图

部分代码如下所示。

```
def askURL(url):  
    head = { "User-Agent": "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.71 Safari/537.36 Core/1.94.186.400 QQBrowser/11.3.5195.400"}  
    request = urllib.request.Request(url, headers=head)  
    html = ""  
    response = urllib.request.urlopen(request)  
    html = response.read().decode("utf-8")  
    return html
```

在获取到响应内容之后，需要从 bs4 中导入 BeautifulSoup 库。该库能够提供许多简单的、Python 的函数来实现处理导航、搜索、修改分析树等，从而实现从网页中抓取数据。BeautifulSoup()函数有两个参数，一个是 URL，另一个是解析器。对于访问 HTML 来讲就是直接使用 bs4 自带的 html.parser。这样便已经成功的获取到所要访问网站的源代码了。但我们想要获取到指定的一些内容还需要进一步的操作。通过使用 BeautifulSoup 库提供的 find\_all()函数就可以指定选择我们想要的所有某标签里的内容。例如本文选择的豆瓣网页中所有标签名为“sidebar-info-wrapper”的<div>标签。这样便

将当前网页中所有名为“sidebar-info-wrapper”的<div>标签都获取到。在此基础上需要使用正则表达式获取到我们想要的最终的内容，此时需要导入 re 库，进行文字匹配。使用 re.compile()函数编译正则表达式，生成一个正则表达式的对象，以供 re 函数使用。最后通过 re.findall()函数获取到我们所需的内容。此时在获取到的内容中有的文本还需要简单清洗，使用 replace()函数和 strip()函数清洗。部分代码如下所示。

```
findName = re.compile(r'<img.*alt="(.*?)"')
soup = BeautifulSoup(askURL(baseurl), "html.parser")
for movie in soup.find_all('div', class_='sidebar-info-wrapper'):
    moviedata = []
    movie = str(movie)
    mname = re.findall(findName, movie)[0]
    mname = mname.replace("\n", "").replace("\r", "")
    moviedata.append(mname)
```

在获取到需要的内容之后我们就需要将其保存到数据库或 Excel 表中，本文采用的是 SQLite 数据库。导入 Python 中的 sqlite 库，先创建一个数据库 movie.db 再创建数据表 movie。然后将每一次循环中添加的列表写入数据库，通过 SQL 语句执行插入操作。部分代码如下。

```
def saveData2DB(commentlist, dbpath):
    conn = sqlite3.connect(dbpath)
    cur = conn.cursor()
    for data in commentlist:
        for index in range(len(data)):
            data[index] = "" + data[index] + ""
            sql = "insert into movie(username, rating, title, short, moviename, director) values(%s) % " % ", ".join(data)
            cur.execute(sql)
            conn.commit()
            cur.close()
            conn.close()
```

操作执行到这里时，一次爬虫的流程就结束了。之后就是让爬虫一直进入循环爬取到本系统所需的内容。其中还需要添加 time.sleep()函数，来控制爬虫访问的时间，如果访问时间过快会遇到爬取不到内容的情况，以及有可能会被限制 IP 等。

在爬取的过程中会发现，豆瓣网站每一部电影的影评有访问数量限制。对于长影评没有访问限制。而对于短评，当用户未登录时只能查看 200 条短评，当登录之后可以查看 600 条，于是需要用爬虫实现登录以获取更多评论。实现登录操作可以通过导入 requests 库，设置 session，发送账号密码等 data 数据。但是豆瓣在登录后还需要手机号



码验证等一系列操作，于是此方法不适用或者需要更复杂的方式来实现。因此，本文采用另一种方法，通过 Selenium 来实现登录操作。Selenium 是一个用于 Web 应用程序的测试工具，它可以直接运行在浏览器上，模拟用户的真实行为。其工作流程如图 4-3 所示。

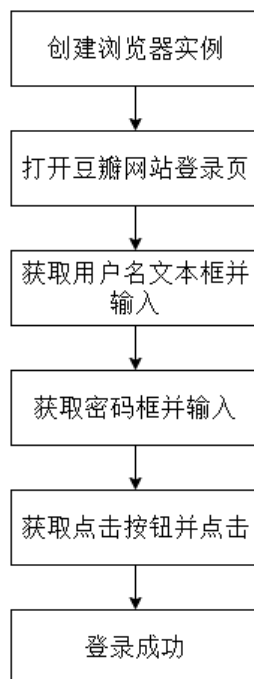


图 4-3 selenium 技术

通过安装浏览器驱动来进行测试，并导入 webdriver 等 Python 库。建立浏览器实例，访问所要访问的网址，并自动输入账号密码完成登录，此时只需要手动完成验证操作，使其登录成功，便可以以登录后的状态访问豆瓣网站并获取源代码得到所需内容。之后的操作与上述爬虫流程操作类似，只是增加一个模拟登录操作。其部分代码如下，其中账号密码部分涉及隐私使用“\*”代替。

```
driver = webdriver.Firefox(executable_path="geckodriver.exe")
driver.get("https://accounts.douban.com/passport/login")
l_button = driver.find_element(By.XPATH, '/html/body/div[1]/div[2]/div[2]/div/div[1]/ul[1]/li[2]')
l_button.click()
username = driver.find_element(By.NAME, 'username')
username.send_keys("*****")
password = driver.find_element(By.NAME, 'password')
password.send_keys("*****")
login_button = driver.find_element(By.XPATH, "/html/body/div[1]/div[2]/div[2]/div/div[2]/div[1]/div[4]/a")
login_button.click()
```

## 4.2 文本预处理

通过 Python 爬虫一共爬取了大概 28000 条评论，其中大约 14000 条评论是积极评论，大约 14000 条评论是消极评论。于是用这些评论进行文本分析。文本情感分析就是通过一些算法去判断一段文本或评论的情感极性，以达到快速地了解文本作者的主观情绪。文本情感分析可以用于舆情监控、信息预测，或用于判断产品的口碑，进而帮助企业改进产品。

本文是对电影评论的分析，主要通过以下流程来实现。首先是加载语料，将爬虫爬取的文本内容从数据库中导出，同时加载停用词。再对导出的语料进行分词处理，同时将停用词去掉，达到清洗的目的。然后将清洗后的分词数据继续进行特征提取等操作，并使用朴素贝叶斯算法进行计算和训练，得到当前语料下的文本情感极性分类器。最后使用语料测试集对训练好的朴素贝叶斯分类器模型进行分类准确性的检验与测试，以验证其性能。

文本分析流程图如图 4-4 所示。

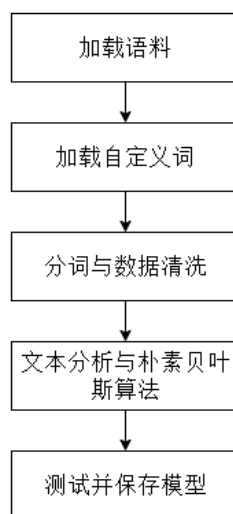


图 4-4 文本分析流程图

### 4.2.1 加载语料

从数据库中导出数据，需要导入 pandas 库，pandas 是一个提供了高效地操作大型数据集所需工具的一个库函数，同时也会使用 numpy 库，它可以存储和处理大型矩阵。使用 pandas 导出数据库为 csv 格式，先使用 `pd.read_sql_query(sql, con)` 函数导出数据，传入 SQL 语句和连接数据库的参数。再使用 `data.to_csv()` 函数保存导出的数据，即完成了数据准备工作。加载语料时通过读取已保存的 csv 文件，将其读取后进行打乱顺序，再将评论和情感分分别存放到两个列表中。

同时还需要加载停用词，将停用词表读取并存储到列表中，以供后续分词使用。部分代码如下。

```

def load_corpus(corpus_path):
    with open(corpus_path, 'r', encoding='utf-8') as f:
        reader = csv.reader(f)
        rows = [row for row in reader]
    review_data = np.array(rows).tolist()
    random.shuffle(review_data)
    comment_list = []
    sentiment_list = []
    for words in review_data:
        comment_list.append(words[2])
        sentiment_list.append(words[1])
    return comment_list, sentiment_list

def load_stopwords(file_path):
    stop_words = []
    with open(file_path, encoding='UTF-8') as words:
        stop_words.extend([i.strip() for i in words.readlines()])
    return stop_words

```

#### 4.2.2 分词和清洗

分词工作需要导入 Python 的 jieba 库，它是一个中文分词组件，可以将一句话分成多个词。在进行分词之前，可以先将用户自定义的词语加入到 jieba 当中，让其对特殊词语分词更精准。使用 jieba.cut() 函数就可以将其分词完成，最后再将其跟停用词表对比，去除停用词，以达到清洗的目的。

### 4.3 基于朴素贝叶斯的情感分析

在概率论中有一种概率叫条件概率，表示为  $P(A|B)$ ，它表示在 B 事件已经发生的条件下，A 事件发生的概率。贝叶斯公式则是在知道  $P(B|A)$  的情况下，计算出  $P(A|B)$ 。贝叶斯公式如下：

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (4-1)$$

而何为朴素贝叶斯，就是在贝叶斯公式的基础上要求属性之间相互条件独立。其中  $P(A|B)$  叫做后验概率， $P(A)$ 、 $P(B)$  为先验概率。因此，使用朴素贝叶斯求解已知某一特征词组  $B(b_1, b_2, b_3, \dots, b_n)$  的前提下属于类别  $A_x$  的条件概率  $P(A_x | b_1, b_2, b_3, \dots, b_n)$ ，由于分母都是一样的，所以简化公式，如下所示：

$$P(A_x | b_1, b_2, b_3, \dots, b_n) = P(b_1, b_2, b_3, \dots, b_n | A_x) P(A_x) \quad (4-2)$$

例如，假设一共有 20 条评论，其中有 12 条是积极评论，8 条是消极评论，那么  $P(\text{积}$

极)=12/20=0.6,  $P(\text{消极})=8/20=0.4$ 。然后从所有的评论中选定了四个关键词作为分类依据,并记录下每一个词在积极评论和消极评论中出现的次数,分别如表 4-1 所示:

表 4-1 评论词出现次数

评论词	在积极评论出现次数	在消极评论出现次数
喜欢	5	1
垃圾	1	6
差劲	0	5
很棒	5	0

那么所有关键词在两种不同评论中出现的概率如表 4-2 所示:

表 4-2 关键词概率

概型	概率
$P(\text{喜欢} \text{积极})$	5/11
$P(\text{垃圾} \text{积极})$	1/11
$P(\text{差劲} \text{积极})$	0/11
$P(\text{很棒} \text{积极})$	5/11
$P(\text{喜欢} \text{消极})$	1/12
$P(\text{垃圾} \text{消极})$	6/12
$P(\text{差劲} \text{消极})$	5/12
$P(\text{很棒} \text{消极})$	0/12

这样就完成了一个简单的朴素贝叶斯分类器,此时就可以测试一段新的评论文本了解情感极性。若这段评论文本中有“喜欢”和“垃圾”两个关键词,则可以通过分类器计算其情感极性。计算结果如下:

$$P(\text{积极})P(\text{喜欢}|\text{积极})P(\text{垃圾}|\text{积极})=0.0248 \quad (4-3)$$

$$P(\text{消极})P(\text{喜欢}|\text{消极})P(\text{垃圾}|\text{消极})=0.0167 \quad (4-4)$$

那么通过计算,这段评论的情感极性可能是积极的概率比可能是消极的概率要高。就可以认为这个评论是积极评论。而“朴素”体现在两个关键词是相互独立的,不会因为出现的顺序或者上下文关系等情况影响。

在了解了朴素贝叶斯算法的理论知识后,本文的系统将采用此方法进行文本分析。生成朴素贝叶斯分类器代码流程如图 4-5 所示。首先将进行分词和清洗后的文本数据按照 1:4 的比例分为测试集和训练集。导入 sklearn 库,该库封装了许多用于机器学习的方法和函数然后对文本提取特征。



图 4-5 朴素贝叶斯分类器生成

使用 `CountVectorizer()` 计算词频，即它能将文本中的词语转换为词频矩阵，再通过 `fit_transform()` 函数计算各个词语出现的次数。然后计算 TF-IDF 值。TF 表示词频，计算方法为：某个词在文本中出现的次数除以文本总词数。其计算公式如公式(4-5)所示。IDF 表示逆文档频率，计算方法为：文档总数除以包含该词的文档数，然后取对数。其计算公式如公式(4-6)所示。TF-IDF 是词频-逆向文件频率，它可以评估一个词对于一个文件集或语料库的其中一个文件的重要程度。TF-IDF 会随着词语出现的次数增加而增加，但同时也会随着它在语料库中出现的次数增加而减少，也就是说，TF-IDF 与 TF 成正相关与 IDF 成反相关关系。因此，TF-IDF 可以帮助我们在文本中找到关键词，并对文本进行分类等操作。其计算公式如公式(4-7)所示。

$$TF = \frac{\text{某个词在文章中出现次数}}{\text{文章的总词数}} \quad (4-5)$$

$$IDF = \log\left(\frac{\text{语料库的文档总数}}{\text{包含该词的文档数} + 1}\right) \quad (4-6)$$

$$TF - IDF = TF * IDF \quad (4-7)$$

因此，使用 TF-IDF 能够计算出文本的核心关键词。将关键词作为朴素贝叶斯分类的特征词。使用 `sklearn` 中的 `TfidfTransformer()` 函数转换为 TF-IDF 矩阵，再用 `fit_transform()` 函数计算 TF-IDF。最后建立朴素贝叶斯分类器，用经过上述处理后的词组与情感得分进行朴素贝叶斯计算得到分类器。将得到的分类器保存为 `pickle` 文件模型，以便后续使用。部分代码如下。

```
comment_list, sentiment_list = load_corpus(file_path)
n = len(comment_list) // 5
train_comment_list, train_sentiment_list = comment_list[n:], sentiment_list[n:]
test_comment_list, test_sentiment_list = comment_list[:n], sentiment_list[:n]
comment_train = [' '.join(comment_to_text(comment)) for comment in train_comment_list]
sentiment_train = train_sentiment_list
review_test = [' '.join(comment_to_text(comment)) for comment in test_comment_list]
sentiment_test = test_sentiment_list
vectorizer = CountVectorizer(max_df=0.8, min_df=3, token_pattern=u'(?u)\b[^\d\\W]\\w+\\b')
tfidftransformer = TfidfTransformer()
tfidf = tfidftransformer.fit_transform(vectorizer.fit_transform(comment_train))
clf = MultinomialNB().fit(tfidf, sentiment_train)
```

#### 4.4 可视化

在完成文本分析之后，为了方便用户使用和视觉效果，做了可视化操作。由于不管是爬虫还是文本分析都是使用 Python 作为主要的编程语言，因此可视化部分也采用 Python 语言和基于 Python 的前端框架。Python 有多种前端框架，如 Django、Flask 等，本文采用 Flask 前端框架。并使用 Web 技术以达到可视化的目的。设计相关的 HTML 网页，将所要展示的内容在网页中展示。在 Flask 框架中使用@app.route()制作相关网址，以达到网页跳转功能。其中还使用 render\_template()函数实现网页展示、跳转和传参等操作，以及使用 ajax 方法来实现前后端的数据传递。本文所提及的系统共设计了五个 HTML 文件，分别为首页、新电影页、电影详情页、文本分析页、更多电影页五个页面，通过 CSS 和 JS 等设计了友好的网页。

##### 4.4.1 Echarts

除了网页的制作，其中可视化技术还包括饼图和词云图，两者为用户直观地展示了电影相关的一些信息。饼图使用了 Echarts 工具，它是一个数据可视化库，能提供用户自己定制的数据可视化图表，包括直线图、饼图、散点图、地图等。Echarts 兼容性好，可定制性强，易于使用和扩展通过 JavaScript 中使用实例。通过将后端的数据传到前端，即可实现可视化饼图。其显示结果如图 4-6 所示，为用户展示了电影评论大致的情感极性分析情况。

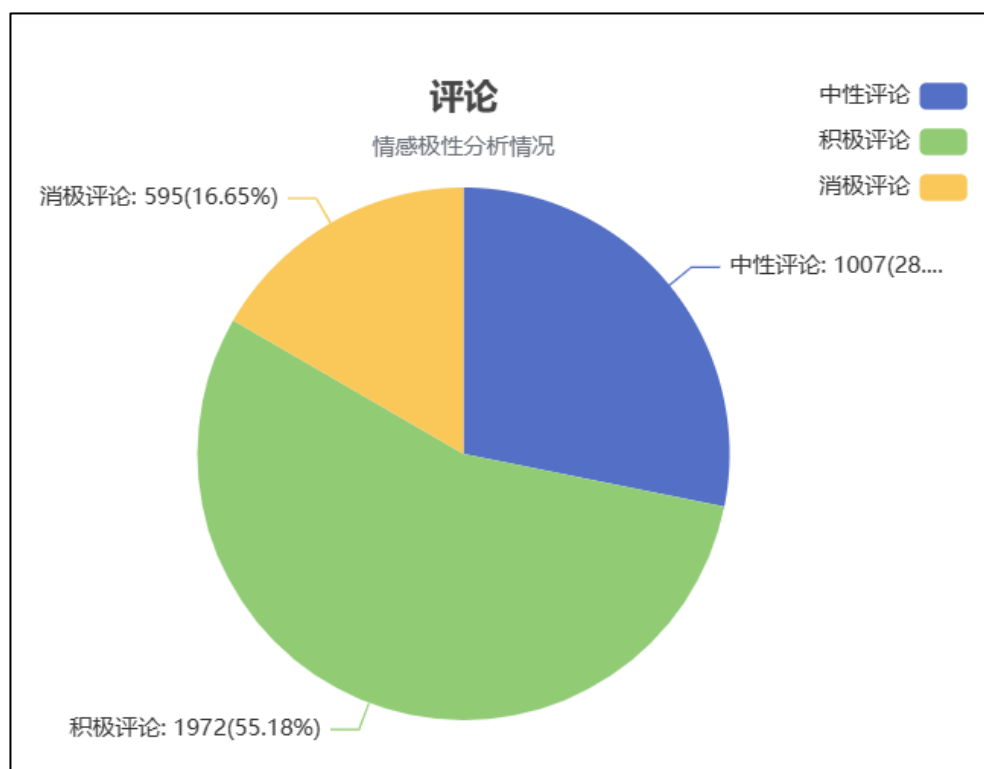


图 4-6 Echarts 可视化饼图

#### 4.4.2 词云图

词云图也是一种可视化手段，它能够根据文本数据的权重展示出不同的大小。词云图样例如图 4-7 所示。



图 4-7 词云图

词语的权重越大，在词云图中展现的比例就越大，因此它可以更加直观的展示出关于电影评论的关键词。所以词云图就是对各种文本数据中出现频率较高的词语予以视觉上的突出，形成关键词云层或关键词渲染，从而达到简化大量文本信息的效果，使用户只要一眼扫过文本就可以领略文本的主旨。像这样以不同的文字杂乱组合在一起，形成

一定形状的图片，不仅能够以非常直观的方式展示出文本的重点，而且形式炫酷，颜色多变，给人一种眼前一亮的感觉。而想要绘制出词云图，需要导入 wordcloud 库、matplotlib 库、PIL 库、numpy 库等相关 Python 库，其中 wordcloud 用来获取词云，matplotlib 用来绘图，PIL 库用来处理图片，numpy 库用来实现矩阵运算。在制作词云图时也需要经过分词和清洗数据去除停用词等操作。然后设置画布大小、字体、背景、dpi 分辨率等操作。部分代码如下所示。

```
wc = WordCloud(  
    background_color='white',  
    height=300,  
    width=450,  
    font_path="msyh.ttc",  
    stopwords=stopwords )  
wc.generate_from_text(string)  
fig =plt.figure(figsize=(24,20),dpi=800)  
plt.imshow(wc)  
plt.axis('off')  
plt.savefig(r'static/img/word{}.jpg'.format(i),bbox_inches='tight', dpi=800,pad_inches=1)
```



## 5 系统测试及结果分析

### 5.1 实验分析与结果

本系统采用朴素贝叶斯机器学习的方法来实现文本聚类和分析。首先是文本预处理，将原本的评论文本进行分词处理，部分结果如图 5-1、图 5-2 所示，其中图 5-1 为原始爬虫爬取下来的评论文本，图 5-2 为经过分词后的文本。

comment
没必要自视过高 更不要自轻自贱我的感受：1 远超预期 2比第一部强 3 格局打开 境界为什么moss说图恒宇是那个变量三更推荐一些相关播客给大家；二刷后的细节补充和更新最从达叔和李老爷子看得出郭帆是个多么有情怀的导演我觉得全片这些动人看点中，片尾那句“如果阿凡达2都可以有8.0？好评夸赞不说了，豆瓣满街都是。这一部非常震撼，中国式科幻美学巅峰 刘慈欣专业户：面壁者马兆、执剑人周喆直带着你我的眼睛，它将是中国电影工业的万里长城沙溢戳吴京那三下，直接戳我心窝里了！我承认，刘培强和韩朵朵的爱情线虽然份额没瑕不掩瑜，迄今为止中国科幻电影的最强音！写了篇这影评谈了谈对影片的几点看法，其中刚看完，都去看，超出所有预期没想到第一时间看完电影，第一时间随手发感想，有那么

图 5-1 原评论文本

cut_comment
没必要 自视过高 更 不要 自轻自贱 我 的 感受：1 远超 预期 2 比 第一部 强 3 格局 打开 境界为什么 moss 说 图恒宇 是 那个 变量 三 更 推荐 一些 相关 播客 给 大家；二刷 后 的 细节 补充 和 更新 最从达叔 和 李老爷子 看得出 郭帆 是 个 多么 有 情怀 的 导演 我 觉得 全片 这些 动人 看点 中，片尾 那句“如果 阿凡达2 都 可以 有 8.0？好评 夸赞 不 说了，豆瓣 满街 都 是。这一部 非常 震撼，中国式 科幻美学 刘慈欣 专业户：面壁者 马兆、执剑人 周喆 直带着 你 我 的 眼睛，它 将 是 中国 电影工业 的 万里长 沙溢 戳 吴京 那 三下，直接 戳 我 心窝 里 了！我 承认，刘培强 和 韩朵朵 的 爱情线 虽然 份额 没 瑕不掩瑜， 迄今 为止 中国 科幻电影 的 最强音！写了 篇 这 影评 谈 了 谈 对 影片 的 几点 看法，其中 刚 看 完，都 去 看，超出 所有 预期 没想到 第一 时间 看 完 电影，第一 时间 随手 发 感想，有 那么

图 5-2 分词后的文本

将数据预处理分词之后的结果进行清洗，去除其中的标点符号以及停用词。其部分清洗之后的文本结果如图 5-3 所示。可以看见，相比于图 5-1、图 5-2，第一行中“不要”一词已被清洗，并且逗号、冒号、感叹号等标点符号也被删除。

clean_comment
没必要 自视过高 更 自轻自贱 感受 1 远超 预期 2 第一部 强 3 格局 打开 境界 打开 4 中国为什么 moss 说 图恒宇 变量 三 更 推荐 相关 播客 二刷 后 细节 补充 更新 看 主创 团队 路演 观众 交流从达叔 李老爷子 看得出 郭帆 多么 情怀 导演 全片 动人 看点 中 片尾 那句 致敬 吴孟达 最让人阿凡达2 都 80 好评 夸赞 不 说 豆瓣 满街 都 一部 震撼 中国式 科幻 美学 巅峰 说 说 打差评 刘慈欣 专业户 面壁者 马兆 执剑人 周喆 直带 眼睛 中国 电影工业 万里长城 第一关 科幻电影沙溢 戳 吴京 三下 戳 心窝 里 承认 刘培强 韩 朵朵 爱情 线 份额 没 很好 嗑 承认 图恒宇 瑕不掩瑜 迄今为止 中国 科幻电影 最强音 写 篇 影评 谈 谈 影片 几点 看法 看法 很多 豆友 讨论刚看完 都 去 看 超出 预期 没想到 第一 时间 看 完 电影 第一 时间 随手 发 感想 质疑 质量 回复

图 5-3 数据清洗

将文本进行预处理之后，则将其按照 4: 1 的比例划分为训练集和数据集，将训练集中的分词进行特征词计算，得到特征值矩阵，部分结果如图 5-4 所示。其中“地下”、“地下城”、“地方”、“地球”等都是特征值。

CYF	CYG	CYH	CYI	CYJ	CYK	CYL	CYM	CYN	CYO	CYP	CYQ	CYR	CYS	CYT	CYU	CYV	CYW
地下	地下城	地为	地位	地区	地去	地名	地址	地外	地带	地往	地心引力	地方	地标	地步	地点	地狱	地球
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7
0	0	0	0	0	0	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	1	0	0	0	0	2
1	2	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	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	4	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4

图 5-4 特征值矩阵

最后测试机器学习部分朴素贝叶斯分类器，查看其学习与测试准确度。一共 28382 条评论文本，按照 1:4 的比例分为测试集和训练集，即训练集文本数量 22706 条，测试集文本数量 5676 条。最后通过朴素贝叶斯训练生成的分类器的准确度为 86.5%。将测试好的朴素贝叶斯分类器模型保存为 Pickle 文件，以供后续数据可视化模块直接调用。分类器训练和测试情况如图 5-5 所示。运行文本分析相关内容，运行的结果如图 5-6 所示。

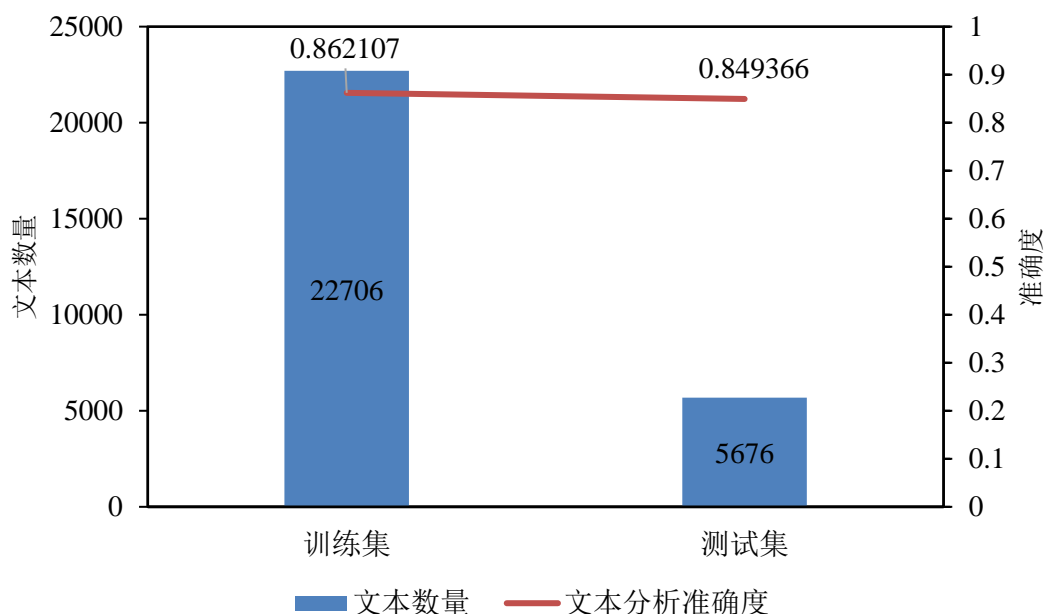


图 5-5 朴素贝叶斯分类结果

训练集数量:	22706
测试集数量:	5676
训练集准确率:	0.8621069320884348
测试集准确率:	0.8493657505285412

图 5-6 朴素贝叶斯分类器测试

在系统测试以及对朴素贝叶斯分类器的测试中，可以发现，对于文本情感极性分析的准确率并没有达到 100%。这其中主要是在利用朴素贝叶斯算法时，对于同一个词语，当其在消极和积极文本中都出现时，其结果在两种情况下的概率都会变得相对较大，可能出现在两种情况下的概率相近，而导致判断准确率下降。而对于要如何解决这部分的问题，则需要更加深度的让机器去学习如何分辨和判断。

同时，在测试的过程中也遇到了一些问题。比如，一开始随机选取了一些电影并爬取其评论，但是最后测试的时候发现，无论怎么训练和测试，最后的测试结果都是好评的概率更高。经过查阅参考文献和学习，了解到需要将积极评论和消极评论的数量匹配一致，这样才能提高准确率。于是另爬取一些差评更多的电影评论，将积极评论数量和消极评论数量大致匹配到 1:1 的效果，即分别有大约 14000 条评论文本。

最后，在完成分类器模型训练后，需要对训练好的朴素贝叶斯分类器进行测试。部分测试结果如图 5-7 所示。

评论	情感得分	测试得分
是怎么做到既不商业又不文艺的	-1	-1
很不错，剧情很舒适，很多小细节小包袱很巧妙	1	1
很纠结这部电影的评分，我承认某些情节确实挺好笑的，但可惜的是在之前上映的电影《不能错过》包了一层费列罗包装的王致和臭豆腐	0	-1
没有什么特别尬的情节，整体质量很高，很搞笑很浪漫，看完了出来还忍不住回味一下	1	1
剧情之尬让人……一言难尽，可以算做是年度烂片了吧	-1	-1
对五一没办法出行的游子来说，整部电影都是献给异地长沙人怀念家乡的礼物，有笑有泪的一部	1	1
电影的节奏很鲜明，关于梦想与现实、爱情与未来的故事在不夜城里上演，导演很有巧思，把几	1	1

图 5-7 情感测试

其中，情感得分是影评用户自己给电影打分，测试得分是通过朴素贝叶斯分类器计算得分。根据结果可以看出，在此部分评论文本中，测试得分跟情感得分几乎相同，说明此模型测试结果有较好的准确性。

## 5.2 系统测试

在完成模型设计以及所有代码编写后，系统便完成了，直接运行 Flask 框架，就可以在本地展示系统完成情况。通过本地网址 <http://127.0.0.1:5000/> 访问，默认使用 5000 端口。运行结果如图 5-8 所示，直接访问显示系统首页相关信息。首页左上角是我本人的 logo 标签，也是网页的一个 logo。右上角分别是首页、新电影页、文本情感分析页和更

多电影页的导航栏，点击便可实现页面跳转。首页中心位置是本系统的相关标题和内容。



图 5-8 系统首页

通过点击右上角的“新电影”标签，实现跳转到新电影页。该页展示最新上映的电影信息，以及这些电影的好评榜。然后主要展示了三部电影的名称、海报和简介，以及更多新上映的另外八部电影的名称和海报。同时将上方导航栏的状态显示到当前页面，即用不同颜色突出显示。该页面的所有电影相关信息都是提前通过爬虫爬取到并保存到数据库中的内容，网页通过查询数据库来显示相关信息。显示结果如图 5-9、图 5-10、图 5-11 所示。



图 5-9 新电影页-好评榜





图 5-10 新电影页-最新上映的电影

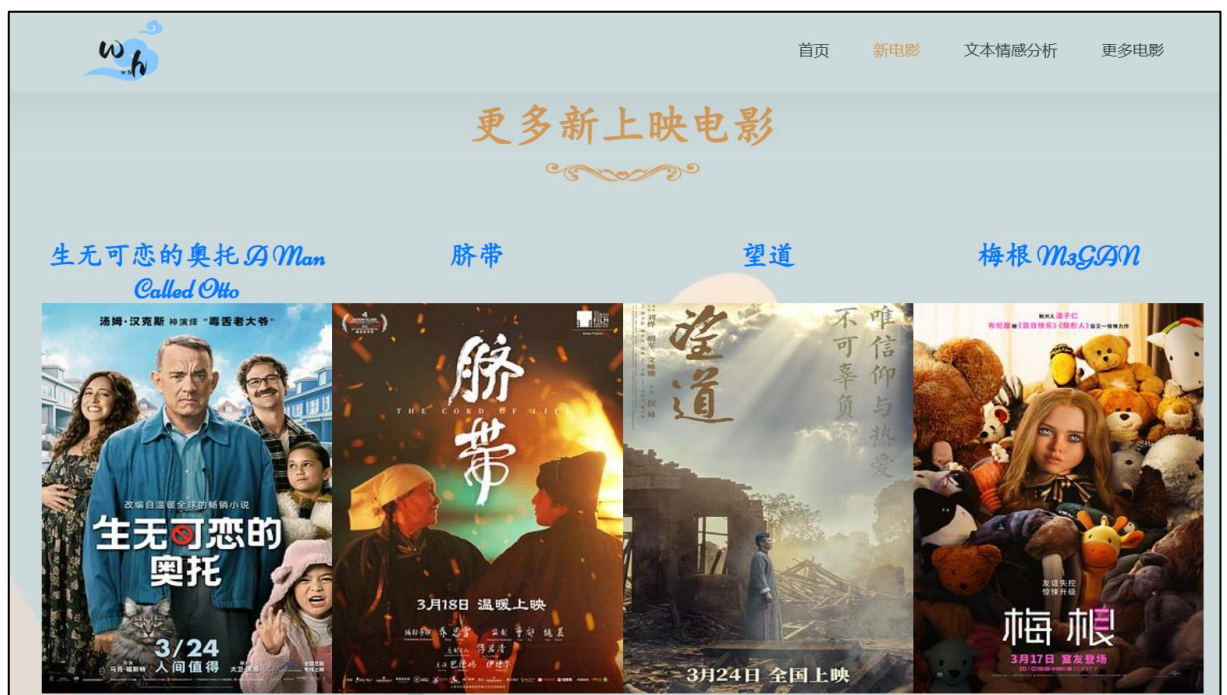


图 5-11 新电影页-更多新上映电影

通过点击电影的名字，便可以跳转到新页面显示该电影的相关信息。主要内容包括电影的词云图，能够直观地展示电影的关键词，让用户一目了然地知道电影的主题。还有电影的评论情况，用饼图展示出积极评论、中性评论和消极评论的数量和占比情况。并在中间显示好评度，好评度包括积极评论和中性评论。最后就是展示影评部分，该部分是通过爬虫爬取到的用户真实评论数据情况。其相关信息展示如图 5-12、图 5-13 所示。

<div><div></div><div><div>首页</div><div>新电影</div><div>文本情感分析</div><div>更多电影</div></div></div>	
<div>电影评论</div>	
电影名称	评论
深海	<p>快醒醒，散场了！本以为那一句话就可以把我从情绪的漩涡中拉离出来，回味却依旧悲伤难以自抑。深海的大饭店是一场美好的梦。这场梦美好到可以治愈一直伤痛。可是梦醒时分，一切都还没有变化。清楚认识到南河的时间痕迹消逝了的那一刻是最悲痛。这个可能不是那么完美的人，却成为了一个人的光。如果可以，宁愿一切都没有发生。晦气是世间都觉得你错了。南河的那——遍遍的“你没有错”直击人心。微笑的面具下是伤痕累累。很多东西，不是那么容易治愈的，原生家庭的痛会伴随一生。长大后的自己某些时刻也有童年的影子。什么时候，不想笑的时候可以笑笑呢？什么样的才是正常人呢？负面情绪像丧气鬼一样，会在低落的时候伸出自己的手让你越陷越深。许多时候，人总是奢望一个救赎。不是每一个人都可以遇到南河，但可以选择成为自己的那一小盏桶灯。暖暖的光慢慢抚慰自己的角落。散场了，撑起伞回家吧。</p>
深海	<p>全篇海水的光影质感，人类、动物的毛发，光影的反射，还有艾尔登法环的女武神猩红辐射，全部到位。这个电影画面是我近几年看过的最最最精细的没有之一，虽然前期剧情不太看的明白，但是当时我就明白不能带逻辑看这个电影，然后我就愉快观影了。画面真的美，细节都非常精细，是非常非常用心的制作，而且到最后剧情一个闷拳给我哭死了😭非常好</p>
深海	<p>是萌萌的海獭与五彩斑斓的亮晶晶！不是。停止萌物海洋的幻想是潜伏着危险与恐惧的无边黑暗。早期海报搜图看到开头那张剧照的时候才想起来。哦。原来去年看到过预告片。所以才在前天做春</p>

然后就是具体用户的评论文本展示，这里随机展示 30 条用户真实的影评，可以通过点击每条评论来查看其情感极性。当点击其中一条评论时，后端则会通过获取当前点击的文本，使用朴素贝叶斯模型计算，最后将计算结果返回到前端，以提示框的形式展

示给用户。比如点击第二条评论“全篇海水的光影质感，人类、动物的毛发，光影的反射，还有艾尔登法环的女武神猩红辐射，全部到位。这个电影画面是我近几年看过的最最最精细的没有之一，虽然前期剧情不太看的明白，但是当时我就明白不能带逻辑看这个电影，然后我就愉快观影了。画面真的美，细节都非常精细，是非常非常用心的制作，而且到最后剧情一个闷拳给我哭死了，非常好”，提示框显示“该评论是 积极 文本！”。其测试结果如图 5-14 所示。



图 5-14 电影评论积极情感极性

例如，在电影《深海》中，有这样一个评论，“前面一些设计好恶心，粘腻的液体，夸张的表情，还有人物特写。一开始进入幻想世界的时候很混乱，人物太多场景嘈杂，看得有些累。最后南河参宿在海上那一段还挺感人的，但是一个儿童版救生圈能撑那么久好神奇。故事较单薄，特效细节做得还是不错的，毛发和皮肤质感值得浅夸一下”，点击该评论，提示框显示“该评论是 消极 文本！”。测试结果如图 5-15 所示。



图 5-15 电影评论消极情感极性

对于一段评论文本是积极评论还是消极评论，对于我们人类而言其实还比较好判断，但是对于中性文本的判断，对于我们人类来说也有可能说不准。而本系统对于中性文本的判断是通过是积极评论的概率和是消极评论的概率，两者之差的绝对值来判断是否属于中性文本。在电影《深海》的评论中选择了这样一条评论，“可能是看过太多表扬它的影评，所以对它的期待值拉得很高很高，感觉电影和我的想象有些许差距。电影的画面色彩，动画质感，配乐都很美很绝。斑斓的深海让人觉得掉进了一场童话梦境，小海獭可爱得让我想伸出手揉一揉。但是故事的剧情有些戛然而止，总让我感觉差点意思，好像再来一点点会更好。很喜欢的情节是南河在抵抗丧气鬼时，转头给参宿画了一个笑脸。此时感觉我就是玻璃背面的参宿，泪流满面。我总在独处时被丧气鬼包围吞噬，感觉世界和我是完全分离的两个个体，感觉这个世界不值得我爱，也没有任何爱我的事物。我的身边没有南河，没有一群小海獭，也没有深海大饭店，只有无法逃离的枷锁和永远到不了的远方。这样的无尽长夜中确实有点星光，而每看到一点，我就会把它捧在手里，带着它走过一段很长的路，戴着它的光芒编织的快乐小狗面具。星光总是渐渐黯淡，下一颗星星会在哪呢。”，点击此评论，提示框显示“该评论是 中性 文本！”，其测试结果如图 5-16 所示。



图 5-16 电影评论中性情感极性

接下来是文本情感分析页，在这个页面中，用户可以将自己想要查询的文本内容输入到文本框中，进行测试，实现查看文本情感极性的功能。用户可以根据自己的需要，分析文本情感极性。用户需要首先在文本框中输入文本内容，然后点击“测试”按钮，在系统完成分析后会把结果显示在下方文本框中。页面详情如图 5-17 所示。



None

测试

请输入你的评论

图 5-17 文本情感分析

例如，当用户输入文本，“情节俗套，剧情狗血且无聊，2023 年居然还能在电影院看到这种剧情，心疼我的电影票钱”，再点击测试，系统会分析出改文本情感极性，并显示在下方文本框中“该评论是 消极 文本！”。测试结果如图 5-18 所示。

情节俗套，剧情狗血且无聊，2023年居然还能在电影院看到这种剧情，心疼我的电影票钱

测试

该评论是 消极 文本！

图 5-18 文本情感分析测试

最后一个模块是更多电影页，在这个页面用户可以根据自己的需要去查询想要了解的电影。因为在前面的新电影页只为用户展示了最近的 11 部电影，所以用户想要了解更多电影则需要在此页面进行搜索。页面如图 5-19 所示。



图 5-19 更多电影

用户在搜索框中输入想要了解的电影名后，点击搜索，系统后端会获取到用户的输入内容，再根据词典查询到对应电影在豆瓣网中的 ID。通过 ID 执行爬虫操作，爬取电影海报和一百条以上的评论，并对评论进行情感极性分析。当爬虫和分析内容都完成后可视化显示在当前页面上，显示电影海报、评论分布饼图和上百条可点击分析的评论。这个过程会加载大概 10 秒钟时间。以搜索电影《长津湖之水门桥》为例，在搜索框中输入电影名，点击搜索，展示结果如图 5-20、图 5-21 所示。



图 5-20 更多电影测试

与新电影页相似，更多电影页中的电影评论也能够通过点击评论，弹出提示框给出该评论的情感极性。



图 5-21 更多电影测试

## 6 总结与展望

### 6.1 总结

随着人们对于电影有更多的追求，以及信息网络的快速发展，在网络上可以看到许许多多不同电影的不同评价。也会有越来越多的人通过在网络中提前了解某部电影的相关信息和评论来判断是否要看这部电影，也可能会为企业和机构提供更好的数据分析服务和决策支持。因此，本文考虑设计一个文本分析系统，以实现文本分析和数据可视化的功能。

本文考虑到基于 Python 的自然语言处理相关的工具应用较为广泛，因此采用 Python 作为系统的开发语言。对于本系统所涉及的爬虫技术，采用了 Python 所提供的 re、BeautifulSoup 等开源库、SQLite 数据库以及 Selenium 工具来实现。对于文本分析功能，采用机器学习中的朴素贝叶斯算法来实现，并通过各种 Python 库来实现机器学习前的准备工作。对于数据可视化技术，采用词云图和 Echarts 饼图等工具实现，其中系统的前端与后端采用了基于 Python 语言的 Flask 框架实现。

本文所设计的系统主要包括 4 个功能，实现了用户查看新电影信息、新电影评论情况、文本情感分析和了解更多电影信息的功能，即对应所设计的网页中新电影页、电影详情页、文本情感分析页和更多电影页。这些功能的实现，能够让每个用户了解到最新的一些电影的相关信息和评论情况，大致得到一个对于某电影的初步判断，了解到该电影评论的情感极性倾向，看自己是否对该电影感兴趣。同时还能让用户自己输入文本来判断所输入的文本情感极性是怎样的，以便于给出适当的评论文本。其中，该系统所设计的影评数据、爬虫技术和文本分析等内容对于一些企业和公司也会有一定的数据分析作用和决策支持作用。

本文最后进行了系统的相关测试，对系统的各模块进行运行和测试。通过对电影影评文本分析，实现了对基于朴素贝叶斯算法的机器学习的分析与测试，验证了本系统的可行性。

### 6.2 展望

本文所设计的系统虽然解决了文本情感分析的相关问题，但仍然存在一定的不足，比如在文本分析模块中使用的基于朴素贝叶斯的机器学习方法可能会导致分析的结果准确率不够高，在更多电影页中使用爬虫技术和文本分析时会消耗大量时间导致页面加载过慢等。基于以上不足，本文将考虑从以下几个方面着手进行优化和改进：

#### （1）采用深度学习或加入情感词典

本文采用的朴素贝叶斯分类算法虽然对于有数据缺失的数据集具有较好的稳定性，但朴素贝叶斯是基于各属性之间相互独立的假设，而该特性是可能不符合实际应用场景的。因此本文考虑在将来采用深度学习或加入情感词典等方法对该系统的文本极性分析功能进行优化，并且增大训练集的数量。

## （2）优化系统功能模块关联逻辑

可以对数据进行预爬取，从而减少用户等待时间，也可以通过使用多线程或优化代码等方法对数据爬取的过程进行优化，提升爬取速度，从而提升系统运行速度以提升用户体验。

另外，本文希望在将来可以与企业合作，接入实际生产过程，提升产品的实际应用价值。

## 参考文献

- [1] 郭丽蓉. 基于 Python 的网络爬虫程序设计[J]. 电子技术与软件工程, 2017(23): 248-249.
- [2] 杜晓旭, 贾小云. 基于 Python 的新浪微博爬虫分析[J]. 软件, 2019, 40(04): 182-185.
- [3] 杨立公, 朱俭, 汤世平. 文本情感分析综述[J]. 计算机应用, 2013, 33(06): 1574-1578+1607.
- [4] 郑诚, 杨希, 张吉赓. 结合情感词典与规则的微博情感极性分类方法[J]. 电脑知识与技术, 2014, 10(13): 3111-3113+3123.
- [5] 高华玲, 张晶. 基于情感词典的酒店评论情感分析与可视化[J]. 软件, 2021, 42(01): 45-47+66.
- [6] 谢松县, 刘博, 王挺. 应用语义关系自动构建情感词典[J]. 国防科技大学学报, 2014, 36(03): 111-115.
- [7] 刘志明, 刘鲁. 基于机器学习的中文微博情感分类实证研究[J]. 计算机工程与应用, 2012, 48(01): 1-4.
- [8] 李春林, 武巾莉. 基于机器学习的白酒板块股评情感分析[J]. 信息技术与信息化, 2021(10): 139-141.
- [9] 葛霓琳, 凡甲甲. 基于朴素贝叶斯和支持向量机的评论情感分析[J]. 计算机与数字工程, 2020, 48(07): 1700-1704.
- [10] 李艳红. 基于机器学习的企业产品评论数据的情感分析研究[J]. 微型电脑应用, 2019, 35(11): 33-35+81.
- [11] 向志华, 邓怡辰. 基于机器学习的文本分类技术研究[J]. 软件, 2019, 40(09): 94-97.
- [12] 黄贤英, 刘广峰, 刘小洋, 阳安志. 基于 word2vec 和双向 LSTM 的情感分类深度模型[J]. 计算机应用研究, 2019, 36(12): 3583-3587+3596.
- [13] 王汝娇, 姬东鸿. 基于卷积神经网络与多特征融合的 Twitter 情感分类方法[J]. 计算机工程, 2018, 44(02): 210-219.
- [14] 梁军, 柴玉梅, 原慧斌, 咎红英, 刘铭. 基于深度学习的微博情感分析[J]. 中文信息学报, 2014, 28(05): 155-161.
- [15] 刘艳梅. 深度学习技术下的中文微博情感的分析与研究[J]. 软件, 2016, 37(05): 22-24.
- [16] 马文, 陈庚, 李昕洁, 苏文伟, 柴焰明, 蒲应明, 曾敬勋, 刘学承. 基于朴素贝叶斯算法的中文评论分类[J]. 计算机应用, 2021, 41(S2): 31-35.
- [17] Ramasamy Madhumathi, Meena Kowshalya A. Information Gain Based Feature Selection for Improved Textual Sentiment Analysis[J]. Wireless Personal Communications, 2022, 125(2): 1203-1219.
- [18] Shailendra Kumar Singh, Manoj Kumar Sachan. SentiVerb system: classification of social media text using sentiment analysis[J]. Multimedia Tools and Applications, 2019, 78(22): 32109-32136.

- [19] Christopher SG Khoo, Sathik Basha Johnkhan. Lexicon-based sentiment analysis: Comparative evaluation of six sentiment lexicons[J]. Journal of Information Science, 2018, 44(4): 491-511.
- [20] Qiu Yue,Zheng Yuchen. Improving box office projections through sentiment analysis: Insights from regularization-based forecast combinations[J]. Economic Modelling, 2023, DOI: 10.1016/J.ECONMOD.2023.106349.
- [21] Zhou Nai,Yao Nianmin,Li Qibin,Zhao Jian,Zhang Yanan. Multi-MCCR: Multiple models regularization for semi-supervised text classification with few labels[J]. Knowledge-Based Systems, 2023, DOI: 10.1016/J.KNOSYS.2023.110588.
- [22] Zhao Pinlong,Han Zefeng,Yin Qing,Li Shuxiao,Wu Ou. Sentiment analysis via dually-born-again network and sample selection[J]. Intelligent Data Analysis, 2020, DOI: 10.3233/IDA-194909.
- [23] Samer Abdulateef Waheeb,Naseer Ahmed Khan,Bolin Chen,Xuequn Shang. Machine Learning Based Sentiment Text Classification for Evaluating Treatment Quality of Discharge Summary [J]. Information, DOI: 10.3390/info11050281.
- [24] Bagla Piyush,Kumar Kuldeep. TA-WHI: Text Analysis of Web-Based Health Information[J]. International Journal of Software Science and Computational Intelligence (IJSSCI),2023,15(1): 1-14.

## 致谢

行文至此，我的本科毕业论文也即将告一段落。在这里，我要真诚地感谢每一位在我完成毕业设计过程中给出帮助的老师、同学、好友。

感谢我的论文指导老师张琳，在我每次对研究方向和论文内容有困惑的时候，都是她及时在我提出问题后给出有指导性的帮助，我也依据她给出的建议继续开展进一步的研究和完善论文写作。她总能在我的毕业设计找不到前进方向的时候指引我正确的方向，也能及时纠正我不合理的研究思路，防止我在错误的方向上越走越远。

我也要感谢我的室友们和朋友们：刘子龙、杨康、吕政泰、沈雨杰、王廷懿、侯浩宇、黑克难、陈晓东、覃文凤、何娴淑、王文慧.....在学习和生活中他们与我互相勉励，督促我按照进度完成毕业论文的设计以及毕业论文的撰写。在我的论文写作过程中，他们在技术上和文本格式上都给予了我很大的帮助。

这篇论文的完成，少不了我生命中最重要的人的支持和帮助。他们就是我的父母。他们是我一生中最坚强、最有爱心、最支持我的人。他们一直默默地支持我，鼓励我奋斗。在我迷茫和挣扎的时候，他们给了我无尽的帮助和理解，让我能够勇往直前。他们的爱让我感到温暖和幸福，使我能够专注于我的研究工作。

最后，我要再次真诚地感谢每一位在我完成毕业设计过程中给予我帮助的人。

或许将来我会再次进入学术圈，也可能会步入工业界，但是无论我的未来如何发展，我都不会忘却我完成本科毕业设计的这段宝贵经历。这段经历教会了我许多，不仅仅是自学能力，更是一种向人生目标奋斗的态度。我相信，这段宝贵的经历会给我未来的人生产生积极的助力效果，伴随着我走完人生。