

# 西安邮电大学

## 毕业设计（论文）

题目： 基于文本情感分析的舆情监测  
系统的设计与实现

学院： 计算机学院

专业： 计算机科学与技术

班级： 计科 1605

学生姓名： 余政

学号： 04161148

导师姓名： 邢高峰 职称：

起止时间： 201X 年 11 月 X 日 至 202X 年 6 月 X 日

## 毕业设计（论文）声明书

本人所提交的毕业论文《基于文本情感分析的舆情监测系统的设计与实现》是本人在指导教师指导下独立研究、写作的成果，论文中所引用他人的文献、数据、图件、资料均已明确标注；对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式注明并表示感谢。

本人完全理解《西安邮电大学本科毕业设计（论文）管理办法》的各项规定并自愿遵守。

本人深知本声明书的法律责任，违规后果由本人承担。

论文作者签名：

日期：        年    月

日

# 西安邮电大学本科毕业设计(论文)选题审批表

申报人	余政		职 称	学 生	学 院	计算机学院	
题目名称	基于文本情感分析的舆情监测系统与实现						
题目来源	科研				教学		其它
题目类型	硬件设计		软件设计		论文		艺术作品
题目性质	应用研究				理论研究		
题目简述	通过抓取微博评论信息，分析文本情感，然后得出情感分析的结论，通过预测得到此微博对民众带来的舆论导向，利用 SnowNLP 模块进行文本分析，Scrapy 框架对数据的采集，Django 对于网站的搭建，以此完成整个项目。						
对学生知识与能力要求	Python 基础，Django 框架，Scrapy 框架，SnowNLP 库						
具体任务以及预期目标	对微博等舆论平台实进行网页分析，找出相应接口，分析 js 代码，解决反爬问题，和相应处理函数，抓取主流的新闻和评论，生成相应数据库文件，存入云端，对评论进行情感分析，生成情感分析词典，构成相应算法和函数，输出情感倾向，生成图表和文档，对舆论走向进行预测，生成相应结论						
时间进度	2019.11.18 ---- 2019.11.30 根据题目，查找资料，设计整体方案，确定系统结构。 2020.01.01 ---- 2020.02.28 将数据存入数据库,对数据进行分析,和确定web 框架。 2020.03.01 ---- 2020.03.30 搭建 web 服务器框架。 2020.05.01 ---- 2020.05.10 完成对爬虫和 web 服务器的实时更新与同步。。 2020.06.01 ---- 2020.06.07 准备各项文档，完成毕业设计论文答辩。						
系（教研室）主任 签字				主管院长 签字			
	年 月 日				年 月 日		

## 西安邮电大学本科毕业设计（论文）开题报告

学生姓名	余政	学号	04161148	专业班级	计科 1605
指导教师	邢高峰	题目	基于文本情感分析的舆情监测系统与实现		
<p>选题目的（为什么选该课题）</p> <p>舆情分析有益于政府组织维护社会秩序和体察民意，由于网络的随意性，某些恶意网民会趁机在网上散播谣言或随意攻击别人，利用舆情分析手段就有助于政府相关部门及时了解事态并采取相应的措施控制和引导舆情，政府部门或相关危机机关可以通过分析大量的文本信息获取舆论走向，但是单凭人工处理这些海量的数据是不切实际的，那么如何让计算机搞笑的从大量的数据中分析用户的情感，得出舆论导向成为了研究热点。舆情分析有较大的市场需求和可行性，因此我本次的课题提出了一个基于情感分析的舆情检测系统。</p>					
<p>前期基础（已学课程、掌握的工具，资料积累、软硬件条件等）</p> <p>已学课程：数据结构，计算机网络基础，数据库，c 语言设计基础</p> <p>掌握的工具：pycharm, navicat, python</p> <p>软硬件条件：软件：python 硬件：windows10 计算机</p>					
<p>要研究和解决的问题（做什么）</p> <p>对微博等舆论平台实进行网页分析，找出相应接口，分析 js 代码，解决反爬问题，和相应处理函数，抓取主流的新闻和评论，生成相应数据库文件，存入云端，对评论进行情感分析，生成情感分析词典，构成相应算法和函数，输出情感倾向，生成图表和文档，对舆论走向进行预测，生成相应结论</p>					
<p>工作思路和方案（怎么做）</p> <p>第一步：首先对微博平台进行分析，分析接口和网页加载情况，是否是 ajax 请求和静态页面请求，构建分布式爬虫爬取热门新闻和评论。</p> <p>第二步：将热门新闻的评论进行存储，构建情感分析系统，将每一句话进行切片，分词，分析词语情感成分，判断句子的情感为负还是正还是中性，存入数据库</p> <p>第三步：将网民评论的情感进行汇聚做成图表，分析网民对于该条新闻大多数持有的态度，和该条舆论对于社会的影响，得到结论。</p>					
<p>指导教师意见</p> <p>余政同学通过查阅大量的资料，对于毕业设计有清晰的认识和完整的解决方案，同意开题。</p> <div style="text-align: right; margin-top: 20px;">             签字：             <div style="display: inline-block; width: 150px; height: 20px; border-bottom: 1px solid black; margin-left: 10px;"></div>             年    月    日         </div>					

# 西安邮电大学毕业设计（论文）成绩评定表

学生姓名	余政		性别	男	学号	04161148		专业 班级	计科 1605	
课题名称										
指导教师 意见	支撑指标点/赋分	3.1/10	3.2/20	3.4/10	4.3/10	8.2/20	10.1/10	12.1/10	12.2/10	合计
	得分									
	<div>指导教师(签字):</div> <div>年 月 日</div>									
评阅 教师 意见	支撑指标点/赋分	3.1/10	3.2/30	3.4/10	4.3/20	10.1/10	12.1/10	12.2/10	合计	
	得分									
	<div>评阅教师(签字):</div> <div>年 月 日</div>									
验收 小组 意见	支撑指标点/赋分	3.1/10	3.2/10	3.4/10	4.2/30	4.3/30	10.1/10	合计		
	得分									
	<div>验收小组(签字):</div> <div>年 月 日</div>									
答辩 小组 意见	支撑指标点/赋分	3.1/10	3.2/10	3.4/10	4.2/10	4.3/10	10.1/50	合计		
	得分									
	<div>答辩小组组长(签字):</div> <div>年 月 日</div>									
学生总评 成绩	评分比例	指导教师(20%)		评阅教师(30%)		验收小组(20%)		答辩小组(30%)		合计
	评分									
	毕业论文(设计)最终等级制成绩（优秀、良好、中等、及格、不及格）									
答辩委员 会意见	<div>学院答辩委员会主任(签字、学院盖章):</div> <div>年 月 日</div>									

## 摘 要

随着时代的发展,互联网迅速发展,我国网民迅速增多。社会网络凭借着开放性、普及性和发散性等收到广大人民群众的喜悦,经过短短几年的发展,进入了全民皆记者的时代。网络的发展使得很多的社交媒体平台应运而生,比如微博,随着微博的发展,每天都有很多的新闻,舆论等事件的发生<sup>[1]</sup>,这些事件就是实时的展示了当下最有意义最热门的事件,并且事情所引发的讨论就成为了民声。因此监测微博的舆论信息具有了重要的意义。本文分析了国内外研究现状以及应用场景。

本系统具有数据采集,数据预处理,情感分析,词云生成等功能。1. 数据采集主要利用了 Scrapy 框架,对 weibo.cn 网站进行抓取,可以通过博主 Oid 或博主的一条博客的 id 进行抓取对应的评论信息。2. 情感分析主要利用 SnowNLP 模块,对评论去除停用词,替换空格等无意义词,分词,情绪判断,关键词提取,断句,预估词频等实现对微博舆论的负面的监测 3. 系统搭建主要利用 Django 框架作为后台管理平台,设计相应数据接口,利用前端页面对数据进行展示,将数据采集和文本情感分析和后台结合。该系统未来可应用于收集民意,引导政府及管理部门完善决策,维持社会稳定。

**关键词:** 社会网络; 舆情监测; 负向舆论; 情感分析;

## ABSTRACT

With the development of The Times, the rapid development of the Internet, China's rapid increase in the number of Internet users. With its openness, universality and divergence, the social network<sup>[2]</sup> has been favored by the masses of the people. After just a few years of development, it has entered the era of journalists for all the people. With the development of the Internet, a lot of social media platforms emerge at the right moment, such as weibo. With the development of weibo, a lot of news, public opinion and other events happen every day. These events show the most meaningful and popular events in real time, and the discussion caused by them becomes the public voice. Therefore, it is of great significance to monitor the public opinion information of microblog. This paper analyzes the research status and application scenarios at home and abroad.

This system has the functions of data collection, data preprocessing, emotion analysis, word cloud generation and so on. 1. Data collection mainly USES the Scrapy framework to grab the website of weibo. The corresponding comment information can be captured by the blogger Oid or a blog id of the blogger. 2. Sentiment analysis mainly use SnowNLP module, to comment to remove the stop, replacing Spaces such as meaningless word, word segmentation, emotional judgment, keyword extraction, pausing, estimate the word frequency and so on to weibo 3. Negative public opinion monitoring system structures, mainly using the Django framework as a background management platform, the design of corresponding data interface, using the front page to display the data, the data acquisition and text sentiment analysis is combined with the background. In the future, the system can be used to collect public opinions, guide the government and administrative departments to improve decision-making and maintain social stability.

**Key words:** social network; Public opinion monitoring; Negative public opinion; Emotional analysis

## 目 录

摘 要.....	3
ABSTRACT.....	7
第一章 绪论.....	1
1.1 背景介绍.....	1
1.2 课题意义.....	1
1.3 国内外研究现状.....	1
1.4 课题所做工作.....	2
1.5 本文章节安排.....	2
第二章 数据处理相关技术.....	3
2.1 数据采集.....	3
2.1.1 爬虫定义.....	3
2.1.2 爬虫框架.....	3
2.2 文本情感分析.....	4
2.2.1 SnowNLP 库[].....	4
2.2.2 SnowNLP 的技术功能原理.....	4
2.2.3 数据预处理.....	4
2.2.4 数据挖掘分析.....	4
2.3 本章小结.....	5
第三章 基于文本情感分析的舆情监测系统设计.....	5
3.1 舆情监测系统的整体架构.....	5
3.2 原始数据的采集.....	6
3.2.1 数据采集系统模型.....	6
3.2.2 数据采集源.....	7
3.2.3 数据采集方案.....	8
3.2.4 反“反爬”方案.....	8
3.3 数据清洗.....	9
3.4 数据的分析和存储.....	10
3.4.1 数据分析.....	10
3.4.2 数据存储.....	10
3.5 本章小结.....	10
第四章 基于文本情感分析的舆情监测系统实现.....	10
4.1 数据采集.....	10
4.1.1 充分调研确定爬虫入口.....	11
4.1.2 伪装身份实现数据采集.....	12
4.2 数据清洗.....	12
4.3 数据可视化.....	12
4.4 本章小结.....	13
第五章 信息系统的部署和测试.....	13
5.1 测试环境的部署.....	13
5.1.1 软件的部署.....	13
5.1.2 硬件部署.....	14
5.2 系统测试.....	14



5.3 本章小结.....	18
<b>第六章 总结与展望.....</b>	<b>18</b>
6.1 项目总结.....	18
6.2 项目展望.....	18
结束语.....	20
致 谢.....	21
参考文献.....	22



## 第一章 绪论

### 1.1 背景介绍

网络是把双刃剑，在这个“全民皆记者”的时代，许多的社会的舆论的事件都始源于网络，以论坛，博客，微博等平台为舆论信息的主要来源<sup>[3]</sup>，并且这些舆论信息的不正确引导会对社会产生巨大的影响。面对互联网的不断发展，我们要以正确的态度来对待它，网路产生的舆论数据的复杂性和异构型等特点，也导致了它的难以控制舆论信心的正确性和传播范围，在我们面对一些热点的话题和焦点的问题的关注很容易导致网络的群体事件或突发事件，所以要实现非常搞笑的网络的监控变得尤为困难。所以如何在庞大的网络的世界里快速、准确的发现有价值的信息，通过设计舆情监测系统，可以帮助政府管理部门及时的发现网络的舆情，及时的引导舆论的方向，稳定社会民众情绪，成为建设现在和谐社会需要待解决的问题。

网络可以让我们随时去了解天下事，但是又因为其开放性和随意性会让一些不法分子乘机而入。作为网络衍生物的网络舆情更是如此，它可以促进政府的民主化建设，但是一些不良信息或者错误的信息会误导判断。为了有效的掌握网络中的信息，各类针对雨落分析的产品应运而生，这类产品大致可以分为4类：一是为政府部门设计的舆情检测系统；二是为高校招生办和研究院设计的舆情系统；三是为了某些艺人危机公关提供的舆情服务；四是为企业提供的舆情产品。各大领域对舆情检测系统的需求证明了舆情分析的重要性。

### 1.2 课题意义

本系统研究目标针对微博平台的舆情评论进行抓取分析，对单条微博评论进行分析后得出情感分析值，优化改良情感分析算法，增加可靠度，使得政府机构可以直接采用此分析结果，对微博中的一些引导社会走向不良风气的微博或者博主进行封禁或删除，具有对微博环境净化和阻止一些居心不良的人利用微博恶意中伤或者发布影响极差的言论的意义。

### 1.3 国内外研究现状

国外舆情分析热点发现研究比较有名的比如美国的 TDT<sup>[4]</sup>(Topic Detection and Tracking)研究项目。该项目的初衷是主要研究出一些算法，能够发现和归纳来自于数据流中的重要信息和内容。用以应对日益严重的互联网信息爆炸问题，对新闻媒体信息流进行新话题的自动识别和已知话题的持续跟踪。热点事件发现

话题与跟踪技术在实际领域中的应用,因此在热点发现的研究中用了很多的 TDT 的技术,主要以新闻语料为研究对象,预料聚类后最终计算热点话题时还需要将相关的报道参数量化来确定最终的结果。

国内的网络舆情分析<sup>[5]</sup>研究呈现出了多元化融合的态势,针对我国目前的国内政治、经济、军事、环境等诸多问题从各个角度做出了相应的研究,且这些研究的内容之间互相联系、交叉融合,并不是独立存在的个体,联系度紧密,如对突发事件的网络舆情分析和引导与意见领袖的引导相互联系,在突发事件中辨析意见领袖的身份等等。各个热点问题相互影响,互相渗透,从一个维度切入的研究内容常常也包含了另一个维度的相关信息,这也是网络舆情分析之所以被称为交叉学科的缘故之一。

#### 1.4 课题所做工作

本课题所作的目标是利用 python 开发的舆情监测系统,利用大学所学知识同时又结合当前行业热门背景和就业岗位等情况,起到学以致用并且可以衔接以后的工作。

对基本的 Django 框架需要会搭建,对目前的微博爬虫接口和一些反爬措施需要有一定的应对措施,对一些常用的 python 库需要一定的了解和使用,对爬虫的框架的使用需要一定的了解。重点实现对用户评论的情感分析,以直观的形式带给使用者更直观的感受。

#### 1.5 本文章节安排

本文根据项目项目的需求一共分为六个章节,层层递进的介绍,具体如下:

第一章绪论整体介绍本项目的项目背景和研究意义,并分析国内外的情况,根据情况来对比本项目实施的可行性和其社会价值性,从而提出了课题的研究目标和未来规划,最后是本文的章节安排。

第二章

## 第二章 数据处理相关技术

### 2.1 数据采集

#### 2.1.1 爬虫定义

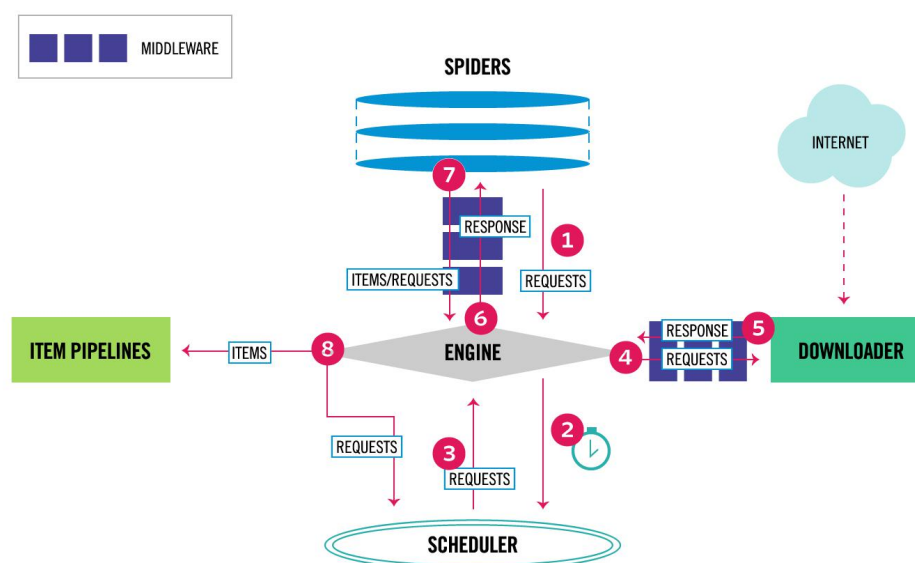
网络爬虫，主要用于收集互联网上的各种资源，它是搜索引擎的重要组成部分，是一个可以自动提取互联网上特定页面内容的程序，一段自动抓取互联网信息的程序。

随着网络的发展，网络上存在的数据越来越庞大，如果仅仅只靠人在庞大的数据中找到需要的数据并且下载下来，是十分消耗时间的，所以使用网络爬虫，它可以代替人们自动地在互联网中进行数据信息的采集与整理。在大数据时代，信息的采集是一项重要的工作，如果单纯靠人力进行信息采集，不仅低效繁琐，搜集的成本也会提高。

#### 2.1.2 爬虫框架

在本舆情监测系统中，网络爬虫技术主要进行的是从微博网站中获取到用户发的微博的所有评论，并且存在数据库中。

在爬虫方面我选择 Python 语言作为项目开发语言，在数据采集方面我使用 Python 中一个比较成熟的爬虫框架 Scrapy<sup>[6]</sup> 框架进行搭建微博爬虫系统。



## 2.2 文本情感分析

### 2.2.1 SnowNLP 库<sup>[7]</sup>

SnowNLP 是一个 python 写的类库，可以方便的处理中文文本内容，是受到了 TextBlob 的启发而写的，由于现在大部分的自然语言处理库基本都是针对英文的，于是写了一个方便处理中文的类库，并且和 TextBlob 不同的是，这里没有用 NLTK，所有的算法都是自己实现的，并且自带了一些训练好的字典。

SnowNLP 具有中文分词、词性标注、情感分析、文本分类、关键词提取、提取摘要、文本分析等功能，利用这些功能可以对文本进行一个简单的分析。

### 2.2.2 SnowNLP 的技术功能原理

SnowNLP 基于朴素贝叶斯算法，在每个不同项目中需要训练不同的情感分析的模型，训练的步骤分为以下几步：

1. 准备正负样本，并分别保存，如正样本保存到 pos.txt，负样本保存到 neg.txt；
2. 利用 snownlp 训练新的模型；
3. 保存好新的模型；

### 2.2.3 数据预处理

爬虫从网页上抓取下来的数据是“脏数据”，需要进行对数据的预处理，主要的处理工作有数据清洗，中文分词，去除停用词等，从微博抓取的新闻评论数据存在较多的干扰信息，比如重复冗余评论，存在 HTML 标签的评论，带有表情的评论等。从 Web 采集下来的新闻评论数据并不干净，使用 Python 正则表达式的方法 re.sub() 匹配去除，通过正则表达式查找和去除 HTML 标签。

中文分词是中文文本情感分析的基础环节，SnowNLP 自带分词功能，所以我直接采用 s.word() 方法进行分词，经过分词后还有存在很多的干扰项，在中文文本中，还存在着某些词汇，虽然使用频率极高但是本身却没有实际的意义，如：“在”“的”“万一”等以一些中文标点符号和键盘符号如：“？”“！”等，这些词汇本身没有什么实际意义，使文本的相似度增加，也增加了文本挖掘的难度。所以需要在分词时过滤掉评论数据中的停用词，排除语料库中的干扰项，同时也能够提高中文分词的准确性。

### 2.2.4 数据挖掘分析

微博评论数据挖掘分析预想方案有三种：

第一种：采用情感字典<sup>[8]</sup>的方式，对文档分词，找出文档中的情感词、否定词以及程度副词，然后判断每个情感词之前是否有否定词及程度副词，将它之前的否定词和程度副词划分为一个组，如果有否定词将情感词的情感权值乘以-1，如果有程度副词就乘以程度副词的程度值，最后所有组的得分加起来，大于0的归于正向，小于0的归于负向。

第二种：采用百度云情感分析接口：  
[http://ai.baidu.com/tech/nlp/sentiment\\_classify](http://ai.baidu.com/tech/nlp/sentiment_classify)  
传入文本参数，利用python中requests模块，使用post请求发出，得到分析结果。

第三种：采用SnowNLP库，准备正负样本对数据集进行训练，得到训练集，对评论的情感进行预测，还可以对评论文本进行分词，统计词频，提炼关键词等功能。

三种方法进行了比较，第一种采用数据字典对分析的可靠性不是很高，预测准确度不高，而第二种方法的预测准确性高，得到的结果多，第三种的功能齐全，开发起来简洁，可以实现分词等功能，最后权衡之下我采用了SnowNLP库进行开发，在未来的项目预期中，我会采用更加可靠的情感分析，使每句话对应情感分析的结果的准确度达到90%之上，对于项目以后和政府部门的对接才能达到预期的目的。

## 2.3 本章小结

本章总体上介绍了本项目中用到的数据采集使用的框架类型和采用的文本情感分析的方法，从基本概念、基本原理和基本使用方法上介绍了几种不同的分析方法的优缺点和场景应用点，为接下来的整个系统设计和实现奠定了一个技术基础。

# 第三章 基于文本情感分析的舆情监测系统设计

## 3.1 舆情监测系统的整体架构

基于文本情感分析的舆情监测系统总体的设计思路是通过对微博平台的数据评论进行数据的采集，获取指定博主（oid）所发布的所有博客下对应的评论，获取系统所需要的基本数据源后，需要对数据进行清洗和抽取，完成对数据的提取和对数据的整合，在经过文本的分析模块的处理后，存入数据库，在可视化操作后，最终由系统搭建的网页上显示出来。

系统总体的设计模型主要采用的是总体规划，分而治之的思想。具体看来就是将系统根据其功能特点分为数据采集模块，数据分析模块，数据可视化模块和系统搭建模块。数据采集模块主要是承担微博信息数据的采集；数据分析模块主要承担数据的清洗，比如分词，去除停用词等，还需要对评论的情感进行分析，将分析的结果交给 Django 然后在网页展示出来；数据可视化主要承担对一条微博的评论进行分割，生成词云，然后在分析的结果上生成可视化的图标，交付后台进行展示；系统搭建主要负责将前三个模块进行整合，在网页中对应应有相应的接口，使得系统更加的紧凑，更加的完整。

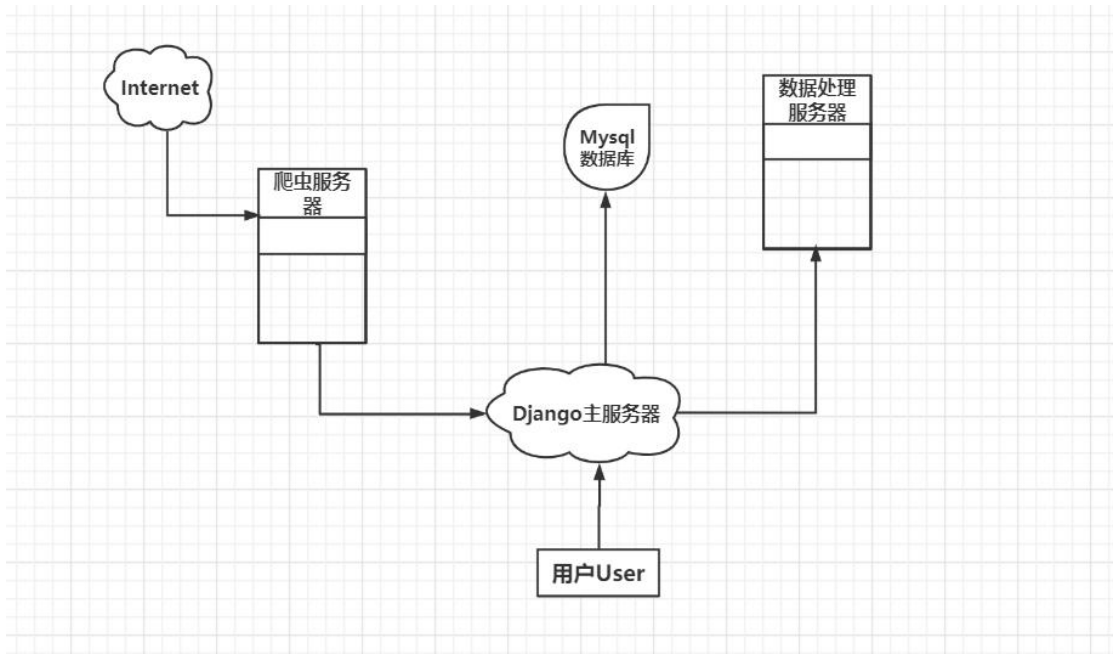


图 3.1 系统整体结构图

## 3.2 原始数据的采集

### 3.2.1 数据采集系统模型

数据采集根据本项目的数据采集对象、数据对象的规模和数据的整体性，数据采集采用服务器集群的模式进行，整个数据的抓取分为了六个部分：主服务器、调度器（Scheduler）、下载器（Downloader）、解析器和存储器。主服务器主要是接受用户的操作指令，然后将命令下发给调度器 Schedule，由调度器操作爬虫进行数据的抓取，主服务器起到对整个系统执行临时任务和数据采集的异常的监控，让整个系统更加的稳定。引擎负责 Spider、ItemPipeline、Downloader、Scheduler 中间的通讯，信号、数据传递等。调度器它负责接受引擎发送过来的 Request 请求，并按照一定的方式进行整理排列，入队，当引擎需要时，交还给引擎。调度器的主体是由 Redis 组成。下载器主要是用来接受调度器传递过来的请求地址，通过相应的协议，在互联网上获取相应的页面的数据。解析器用来接受下载器下



载下来的页面,然后对下载下来的页面进行解析,得到开发者需要得到的数据(此处数据是脏数据)以及请求过来的地址,将解析之后的数据交给存储器,将获取到的请求地址重新交给调度器然后存入调度器的队列。系统模型示意图如图所示 :

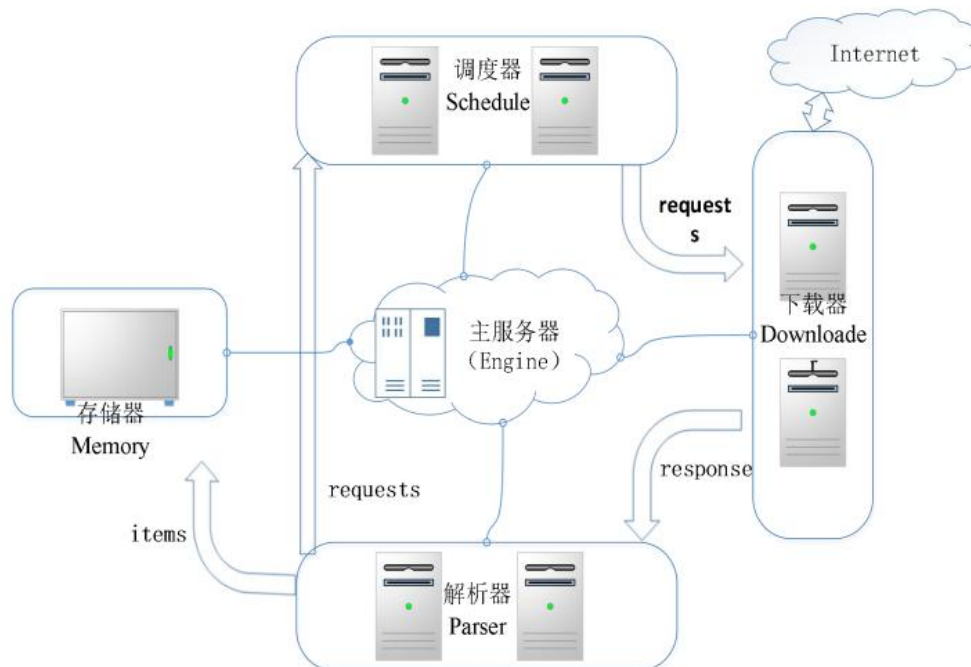


图 3.2 数据采集系统模式示意图

从系统模型示意图可知,首先是主服务器引擎得到命令,产生数据抓取任务,产生一个抓取任务的起始的请求,然后将抓取的任务交付给调度器 Schedule,调度器将下载任务存入请求队列中,然后调度器从请求队列中取出 request 交给 Downloade,Downloade 接收到调度器给的请求任务后,请求网站目标,经过中间件的一些爬虫的配置和一些防反爬的配置后,将网页上的数据获取下来交给解析器 Parser,Parser 对页面的数据进行解析,Parser 需要按照项目的需求进行配置,将需要得到的数据解析出来后交给存储器,需要在网页中继续深度抓取的页面的地址解析出来交给调度器,由调度器存入队列等待下载器空闲之后,由下载器再一次请求,在交给存储器后,存储器将数据按照要求存储的数据库中或者本地文件中。重复上面的步骤,直到将所有需要的数据采集下来,在数据采集的过程中,主服务器一直处于运行状态,它需要监控其他的模块的运行状态,保证数据采集系统的稳定运行。

### 3.2.2 数据采集源

微博是指一种基于用户关系信息分享、传播以及获取的通过关注机制分享简短实时信息的广播式的社交媒体、网络平台,允许用户通过 Web、Wap、Mail、App、IM、SMS 以及用户可以通过 PC、手机等多种移动终端接入,以文字、图

片、视频等多媒体形式，实现信息的即时分享、传播互动。

由于微博的常用性，普遍性，很多的人都喜欢在微博上发表一些自己的言论和一些看法，所以本项目的目标数据网站就选定了微博作为目标，在对微博网站反爬情况的调研中发现微博一共有三个网址 m.weibo.com,weibo.com,weibo.cn。在抓取的难易程度上我选择了比较容易的 weibo.cn 这个网站作为我的数据采集源。

### 3.2.3 数据采集方案

首先，通过对“微博”数据的抓取，获取单条微博的文本内容，发布时间，点赞数，评论信息等作为网站建设的基础。然后在获取到博主的 oid 后，使用爬虫框架将博主的所有发布的博客的信息抓取出来。

名	类型	长度	小数点	允许空值 (	
c_id	varchar	50	0	<input type="checkbox"/>	1
c_created_at	datetime	0	0	<input type="checkbox"/>	
c_source	longtext	0	0	<input type="checkbox"/>	
c_text	longtext	0	0	<input type="checkbox"/>	
c_like_num	int	11	0	<input type="checkbox"/>	
c_userid	varchar	50	0	<input type="checkbox"/>	
c_user_name	varchar	300	0	<input type="checkbox"/>	
C_profile_image_url	longtext	0	0	<input type="checkbox"/>	
C_profile_url	longtext	0	0	<input type="checkbox"/>	
CommentWeiboInfo_id	varchar	50	0	<input type="checkbox"/>	

图 3.3 数据采集数据库表结构

### 3.2.4 反“反爬”方案

如今利用最多的反爬措施<sup>[9]</sup>主要是监测一段时间内某一个 IP 或者 Cookie 请求的次数来判断，称为 IP 封锁，面对常见的反爬措施主要有 ajax 动态加载，js 数据加密，验证码等，对抗某一种反爬措施一般的方法如下：

#### 1.IP 封锁

因为会对用户产生误伤，所以网站一般不会对用户的 IP 进行长时间的封锁。

解决方案：

- (1) 修改程序的访问频率
- (2) 使用 IP 代理的方式来对网站进行爬取

#### 2.协议头

绝大多数网站，访问时会判断访问来源。

解决方案：

- (1) 访问时添加协议头

### 3.验证码

当用户请求频率过高的时候，有些网站就会触发验证码验证机制。

解决方案：

接入打码 API，例如云打码。

### 4.需要登录

有些网站需要用户登录之后才能够获取页面中的信息，那么这种防护能非常有效的防止数据被大批量的被爬取。

解决方案：

- (1) 小数据量进行爬取(模拟登录后再去爬取，或者使用 cookies 直接进行爬取)
- (2) 申请诸多的账号去养这些号，然后登录，或者获得 cookies 进行爬取。

### 5、动态页面的爬取

有一些网站的数据和图片是用 JS 代码动态生成的，那么服务器端，就会通过判断该用户是否访问了这些资源来判断是否爬虫。

解决方案：

- (1) 使用 selenium 自动化测试工具，将网页在 selenium 上将 ajax 页面加载完毕，然后返回网页的数据

## 3.3 数据清洗

微博数据的清洗主要是分为两个大部分，分别是博主的基本信息表 3.3.1 和博客评论的主要信息表 3.3.2：


名	类型	长度	小数点	允许空值 (	
► _id	varchar	20	0	<input type="checkbox"/>	 1
Image	longtext	0	0	<input type="checkbox"/>	
NickName	varchar	30	0	<input type="checkbox"/>	
Gender	varchar	6	0	<input type="checkbox"/>	
Province	varchar	30	0	<input type="checkbox"/>	
City	varchar	30	0	<input type="checkbox"/>	
BriefIntroduction	varchar	500	0	<input type="checkbox"/>	
Birthday	date	0	0	<input checked="" type="checkbox"/>	
Num_Tweets	int	11	0	<input type="checkbox"/>	

表 3.3.1 博主信息数据库表

名	类型	长度	小数点	允许空值 (	
►_id	varchar	50	0	<input type="checkbox"/>	 1
Content	longtext	0	0	<input type="checkbox"/>	
PubTime	datetime	0	0	<input type="checkbox"/>	
Co_ordinates	varchar	300	0	<input type="checkbox"/>	
Tools	varchar	300	0	<input type="checkbox"/>	
Like	int	11	0	<input type="checkbox"/>	
Comment	int	11	0	<input type="checkbox"/>	
Transfer	int	11	0	<input type="checkbox"/>	
UserInfo_id	varchar	20	0	<input type="checkbox"/>	

表 3.3.2 博客评论数据库表

### 3.4 数据的分析和存储

#### 3.4.1 数据分析

对于数据分析使用的是 SnowNLP 库进行分词，统计词频，提取关键词，情感分析，需要将博客的标题和对应的评论进行分析，得到情感值和关键字，最后生成词云和数据柱状图的数据。

#### 3.4.2 数据存储

本项目的数据存储使用的是 Mysql5.7.26 版本。图形数据主要存储用户头像，词云生成的图片，主要在 spiderapi\_image 和 spiderapi\_iserinfo 中，文本形式的数据主要存储在 spiderapi\_commentinfo，spiderapi\_commentweiboinfo，spiderapi\_target，spiderapi\_tweetsinfo 中。

### 3.5 本章小结

本章主要从系统的整体架构，原始的数据采集和数据的分析和存储的系统这三部分进行介绍，从而构建起系统的整体架构的模型框架，和预期可达到的可以投放市场的目标产品的情感分析的方式。

## 第四章 基于文本情感分析的舆情监测系统实现

### 4.1 数据采集

根据前面的设计需求，爬虫的主体是微博平台，对于微博这个大平台来说，数据是非常重要的，那么在这么一个大型的网站上来说，应该如何采集它的数据呢？

### 4.1.1 充分调研确定爬虫入口

在对微博进行数据采集的时候，首先要对该网站进行调研，找到数据对应的接口，或者找到合适的参数才能将数据获取下来，在对微博进行调研的时候，一共发现了微博的三个网址 m.weibo.cn,weibo.cn 和 weibo.com，weibo.com。m.weibo.cn 用的是 ajax 请求的接口作为数据的获取方式

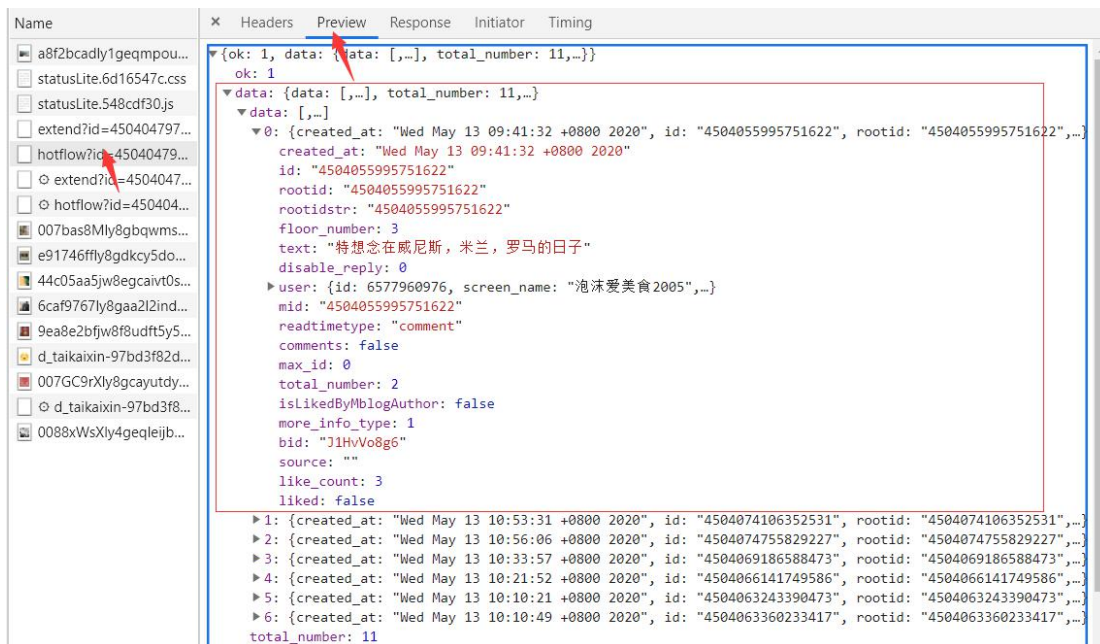


图 4.1.1 ajax 请求

Weibo.cn 中在源码中直接会有数据：

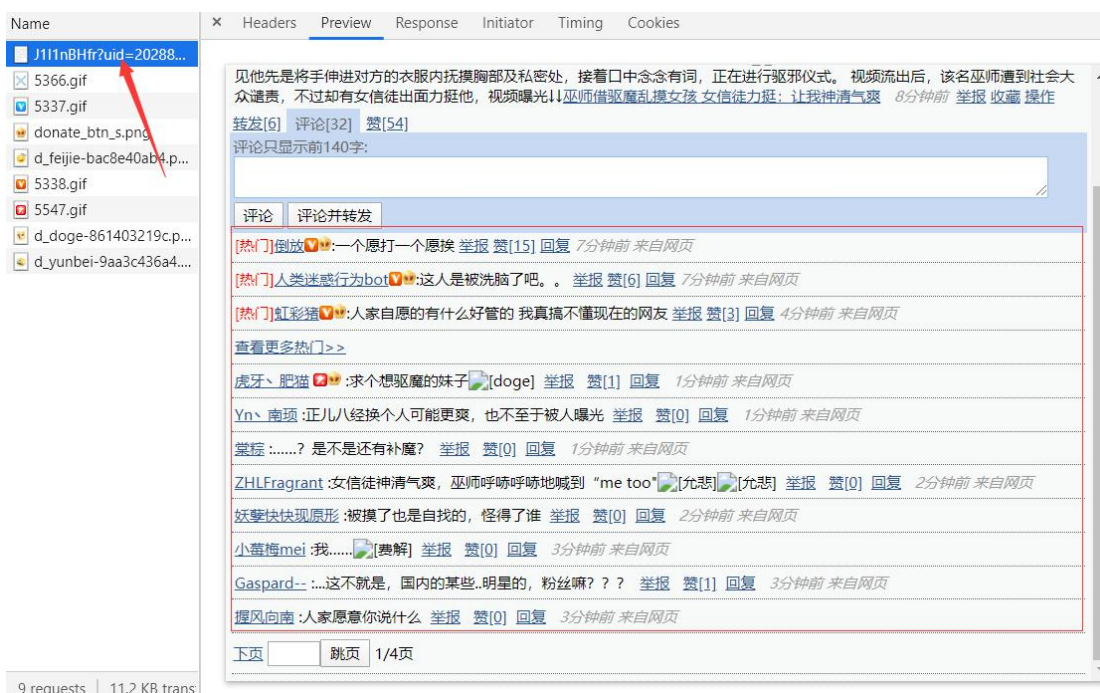


图 4.1.2 weibo.cn 数据图



Weibo.com 的数据比较难以获取,反爬措施相比于前两个难度大了很多,在多次调研后决定放弃使用 weibo.com 的抓取,在后续的抓取中,发现 weibo.cn 和 m.weibo.cn 的 cookie 用的是同一个,在前面的两个比较数据采集难度后,最后决定是用 m.weibo.cn 的网站作为抓取的原网址。

#### 4.1.2 伪装身份实现数据采集

通过 4.1.1 找到数据抓取的接口 <https://weibo.cn/%s/info>,将 %s 参数换为 oid 即可获取到该博主的信息数据,调研中发现该网站需要模拟 User-agent 和模拟登陆的 cookie,所以需要先 will cookie 添加入请求当中,cookie 保存在数据库中, user-agent 获取了常用的所有的浏览器的 agent,每次请求时随机使用一条和 requests 请求发送出去,实现抓取。

#### 4.2 数据清洗

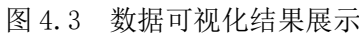
数据清洗主要是采集到的脏数据进行解析、格式化和提取有效的信息。

页面的解析主要靠 Xpath 进行解析完成的,同时搭配正则表达式解析一些不符合正规的 html 布局代码的数据,并将最后的数据进行 json 化,方便后期的存储,由于数据在抓取下来后,有可能有很多的重叠的数据,所以要对数据进行清洗,剔除重复数据,达到清洗的目的。

#### 4.3 数据可视化

通过数据分析和应用,主要用在网站建设后词云的生成和情感折线图和敏感率统计图。

数据可视化是利用前端和后台实现的,在分析结束后将结果存入数据库,在前端页面需要展示词云,在后台数据进行获取,从而直接提高界面的友好度。



本章主要是从实现的角度来介绍系统的实现过程，重点阐述了采集数据的过程，在本系统中，采集也是项目中的重点，也是最复杂的地方，阐释了数据分析后的结果的展示，数据可视化的过程。

## 5.1 测试环境的部署

软件系统的配置主要有 Django 配置<sup>[10]</sup>、Scrapy 配置、Mysql 配置、Redis 配置、Python 安装配置

表 5.1 软件安装部署要求

软件名称	软件版本	软件位数
Django	2.2.10	64bit
Scrapy	1.6.0	64bit
Mysql	5.7.26	64bit
Redis	2.6	64bit
Python	3.7	64bit

（附注：基于系统的开发语言值 Python，系统运行所需要的 Python3 开发库的安装文件在 requirements.txt 参见附录 A）

### 5.1.2 硬件部署

本系统部署在 windows10 操作系统上，物理机内存 8G，硬盘 500G。运行 IP 地址 127.0.0.1（localhost）。

### 5.2 系统测试

本系统定性为测试版系统，下面分别从源数据、数据存储和前端效果展示的测试结果。

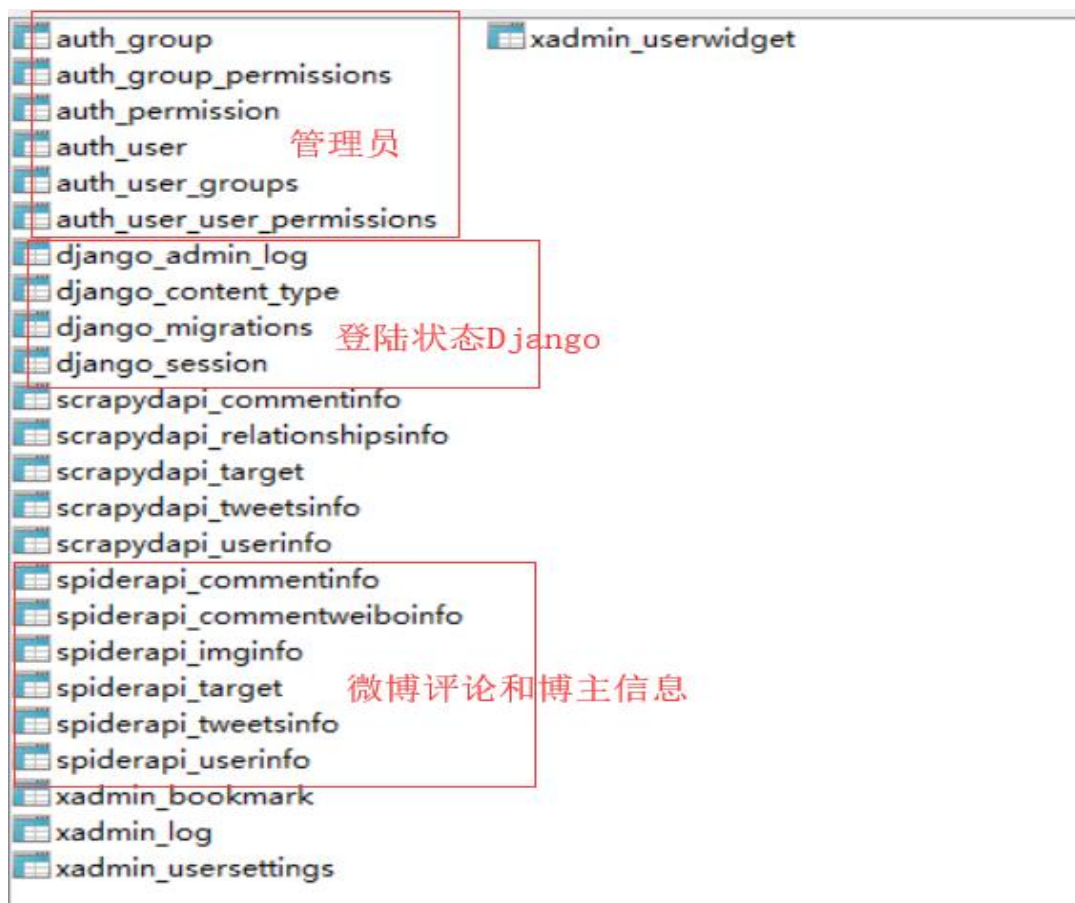


图 5.2.1 数据存储及其效果图



_id	Image	NickName	Gender	Province	City	BriefIntroduction	Birthday
1642512402	(BLOB)	中国新闻周刊	男	北京		中国新闻周刊官方微博。定	(Null)
1642591402	(BLOB)	新浪娱乐	女	北京	海淀区	新浪娱乐	(Null)
1944983397	(BLOB)	梦里诗书	男	湖北	武汉	独立影评人，特约供稿于全	(Null)
1974576991	(BLOB)	环球时报	男	北京		报道多元世界 解读复杂中	(Null)
2319877413	(BLOB)	笑趴了	男	广东		我只是喜欢一个将快乐收集	(Null)
2713866045	(BLOB)	主播金三秒	男	浙江		【金天排行榜】创始人 幽默	(Null)
2803301701	(BLOB)	人民日报	男	北京		人民日报法人微博。参与；	(Null)
5407847973	(BLOB)	神最右最搞笑	男	安徽		关注右哥的人都找到对象了	(Null)

图 5.2.2 博主信息数据库效果图

图 5.2.1 是 Mysql 中数据存储及其效果图，其中生成的词云图片存储在 imginfo 当中，图 5.2.2 是博主的信息存储的结构。



图 5.2.3 微博用户情感分析系统首页图

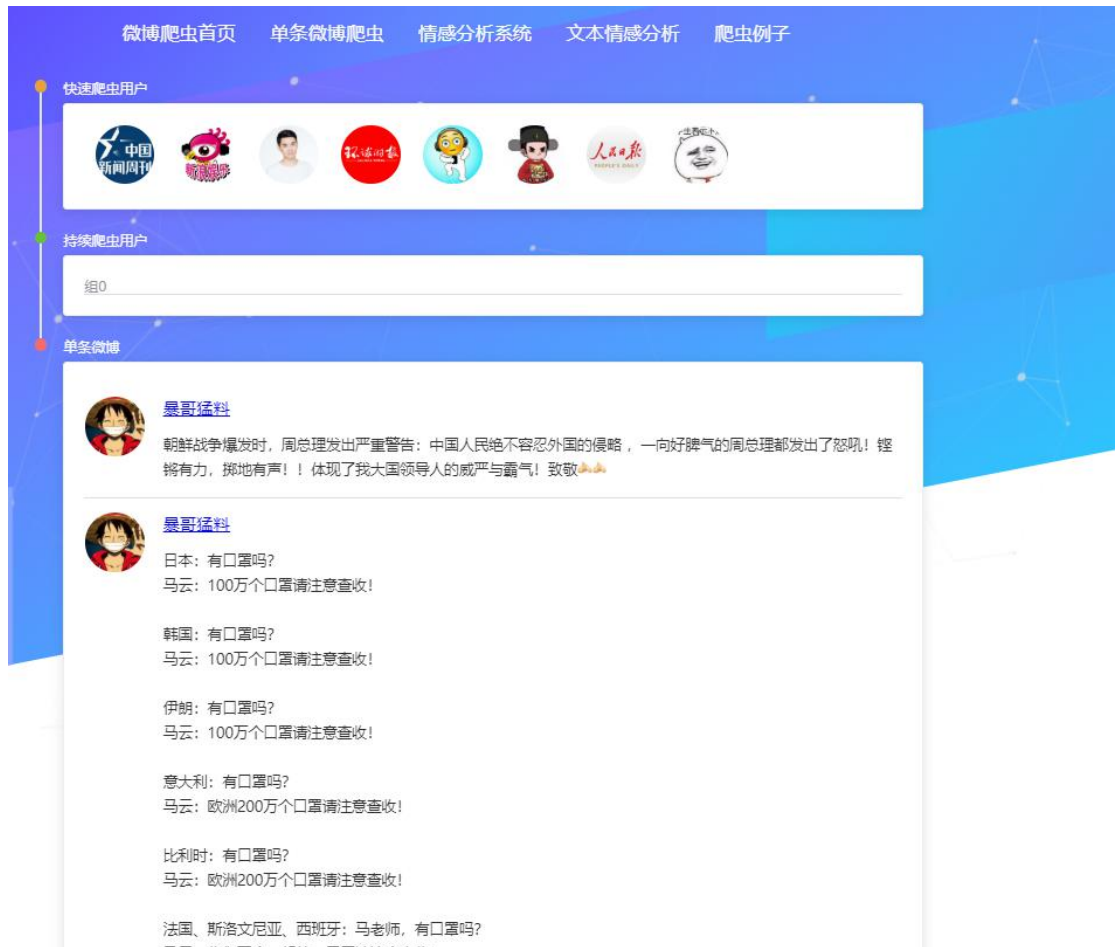
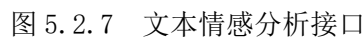
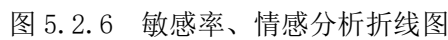


图 5.2.4 爬虫例子截图

上图表述已经爬取的数据的结果展示，快速爬虫用户是指对博主 oid 数据进行爬取，单挑微博是对单个微博的数据的抓取。



图 5.2.5 词云生成展示图



17

表中就会自动分析出文本的情感值。

### 5.3 本章小结

本章主要介绍了对系统的硬件部署和软件部署，以及对系统的测试截图，从测试结果上来看，本系统的整体达到预期的目标。

## 第六章 总结与展望

### 6.1 项目总结

本项目的设计灵感来源于当前最火热的人工智能，在人工智能的情感分析方面，我选择了微博作为项目的源数据，因为评论所带来的社会影响十分严重，所以想到了这个项目在对舆论数据进行分析，为净化社会不良气息做出一定的贡献。在整个毕业设计的过程中主要的工作总结为以下几点：

（1）课题的社会价值的调研和课题的最终确定。项目在准备阶段，进行的项目的社会的调研，在参考了国内外研究现状后确定了题目，并最终确定题目的方向，提出了基于文本情感分析的舆情监测系统的题目。

（2）数据采集和处理的技术的研究和学习。在项目的课题的知识储备阶段，学习到了 python 对于爬虫方向开发的便捷性，通过爬虫技术与数据源平台的对抗，获取系统建设最初的数据源来达到数据采集的目的。

（3）数据分析到数据可视化，数据分析的三种方法的选择，情感字典的构建、百度分析接口和 SnowNLP 分析库的应用，在数据可视化词云的生成，到前端和后台 api 之间的数据交互。

（4）服务器的搭建，利用 Django 框架将所有的模块链接到一起，使得项目的整体化，使用户操作简单，网页设计接口和前端的可视化，最终整个系统的搭建也基本达到了预期的设计和测试要求。

### 6.2 项目展望

本课题基于文本情感分析的舆情监测系统包括数据的采集，数据的分析和挖掘，数据的可视化操作，系统的整体搭建，还有很多可以优化的地方。

（1）数据的采集。整体系统的微博数据应该是有时效性，在一篇新的微博发出来的时候就应该去采集微博的评论然后做出分析，展示出来，所以需要持续做持续性爬虫。

（2）数据的分析与挖掘，在采用了 SnowNLP 库后，分析的结果没有百度 api 那么准确，所以在项目的展望中，对数据的分析采用人工之能，利用优化算

法，使得评论的情感的判断的准确率达到百分之 95 以上，之后才可以将本项目投入到政府机构使用，才不会使得很多的微博被误判。

（3）在对项目的预警中还没有实现，所以对博主的评论的判断，如果消极的评论占大多数，并且情感负面极为严重，就需要做出预警，发送邮箱到管理员或者在网页上以红色显示，做出预警。

## 结束语

本课题基于文本情感分析的舆情监测系统通过采集微博的博主的博客和博客的评论信息，对数据进行分析 and 处理，获取有效的数据，进而在网页上进行展示。

就完成情况而言，项目从数据的采集和最终的可视化展示来看，目前已经通过测试，目前的使用价值只是对于个人，要想投入使用，需要优化其文本情感分析算法，人工智能算法，加强训练集强度，使得分析的准确率提高。

回顾过去，展望未来，对于项目的不足和技术的完善，我将在工作过程中不断的优化和提升，学以致用，希望可以将本项目投入政府机构，为微博平台的舆论净化提供一些微薄之力。

## 致 谢

感谢我的导师邢高峰，在一开始的选题中，到后面的论文和答辩中，老师兢兢业业，帮我克服了很多的困难；

感谢我的室友们，从遥远的家来到这个陌生的城市里，是你们和我共同维系着彼此之间兄弟般的感情，维系着寝室那份家的融洽。

感谢我的爸爸妈妈，焉得谖草，言树之背，养育之恩，无以回报，你们永远健康快乐是我最大的心愿。

在论文即将完成之际，我的情绪无法平静，从开始进入课题到论文的顺利完成，有多少可敬的师长、同学、朋友给了我无言的帮忙。

## 参考文献

- [1] 洪小娟,宗江燕,于建坤,黄卫东. 网络舆情监测系统的分析与设计[J]. 软件工程,2019,22(08):37-39+13.
- [2] Maha Heikal,Marwan Torki,Nagwa El-Makky. Sentiment Analysis of Arabic Tweets using Deep Learning[J]. Procedia Computer Science,2018,142.
- [3] 李丽华,胡小龙. 基于深度学习的文本情感分析[J]. 湖北大学学报(自然科学版),2020,42(02):142-149.
- [4] Zhou Jun,Xu Haimin,Zeng Hong,Ma Hongjun,Yu Jingjing,Liu Xia,Sun Zhulei,Zhou Luting,Zheng Saifang,Wang Xue,Wang Anran,Wang Chaofu. Expression of TdT in Myoepithelial Cells: Investigation in Breasts, Sweat Glands, and Salivary Lesions Emphasizing the Never-Documented Immunohistochemical Findings. [J]. International journal of surgical pathology,2020.
- [5] 唐存琛,毕翔. 国内外网络舆情分析比较研究[J]. 西南民族大学学报(人文社科版),2018,39(09):141-147.
- [6] 郭锋锋. 基于python的网络爬虫研究[J]. 佳木斯大学学报(自然科学版),2020,38(02):62-65.
- [7] 曾小芹,余宏. 基于Python的商品评论文本情感分析[J]. 电脑知识与技术,2020,16(08):181-183.
- [8] 孙本旺,田芳. 藏文情感词典的构建及微博情感计算研究[J]. 计算机技术与发展,2018,28(11):212-216.
- [9] 邹科文,李达,邓婷敏,李嘉振,陈义明. 网络爬虫针对“反爬”网站的爬取策略研究[J]. 电脑知识与技术,2016,12(07):61-63.
- [10] 白昌盛. 基于 Django 的 Python Web 开发[J]. 信息与电脑(理论版),2019,31(24):37-40.



## 附录 A

requirements.txt 文件:

astroid==2.1.0
attrs==19.1.0
Automat==0.7.0
autopep8==1.4.3
backports.csv==1.0.7
beautifulsoup4==4.7.1
bs4==0.0.1
certifi==2019.3.9
cffi==1.12.2
chardet==3.0.4
colorama==0.4.1
constantly==15.1.0
cryptography==2.6.1
cssselect==1.0.3
cycler==0.10.0
defusedxml==0.5.0
diff-match-patch==20181111
Django==2.1.10
django-cors-headers==2.4.0
django-crispy-forms==1.7.2
django-formtools==2.1
django-import-export==1.2.0
et-xmlfile==1.0.1
future==0.17.1
httplib2==0.12.1
hyperlink==18.0.0
idna==2.8
incremental==17.5.0
isort==4.3.4
jdcal==1.4
jieba==0.39
js2xml==0.3.1
kiwisolver==1.0.1
lazy-object-proxy==1.3.1
lxml==4.3.3
matplotlib==3.0.3

mccabe==0.6.1
numpy==1.16.2
odfpy==1.4.0
openpyxl==2.6.1
parsel==1.5.1
Pillow==6.0.0
ply==3.11
pyasn1==0.4.5
pyasn1-modules==0.2.4
pycodestyle==2.5.0
pycparser==2.19
PyDispatcher==2.0.5
PyHamcrest==1.9.0
pylint==2.2.2
pylint-django==2.0.5
pylint-plugin-utils==0.4
PyMySQL==0.9.3
pyOpenSSL==19.0.0
pyparsing==2.4.0
python-dateutil==2.8.0
pytz==2018.9
pywin32==224
PyYAML==5.1
queuelib==1.5.0
requests==2.21.0
Scrapy==1.6.0
scrapy-djangoitem==1.1.1
scrapyd==1.2.0
scrapyd-client==1.1.0
service-identity==18.1.0
six==1.12.0
slimit==0.8.1
soupsieve==1.9
tablib==0.13.0
Twisted==19.2.1
typed-ast==1.3.1
urllib3==1.24.1
w3lib==1.20.0
wordcloud==1.5.0
wrapt==1.11.1
xlrd==1.2.0
xlwt==1.3.0

---

zope.interface==4.6.0