摘 要

随着互联网技术的不断发展，社交网络在人们的日常生活中扮演着越来越重要的角色，同时也在改变着信息的传播方式，从原来的平面媒体与电视广播相结合到现在多元化平台的改变。人们获取热门信息和自己感兴趣的信息的方式也与社交网络变得越来越密不可分。伴随而来的是通过分析大量的用户数据对用户的各种行为进行深度挖掘，从而优化信息传递效率，节约人们从海量信息中提取对自身有价值信息的时间，并挖掘其潜在的巨大的商业价值。国外社交网络Facebook与Twitter的巨大成功也在很大程度上促进了国内社交网络平台的发展。

本文以国内的热门社交网络平台新浪微博的用户信息与微博信息为研究对象，使用Elasticsearch和Kibana来实现针对微博树洞的数据分析。之所以选择微博树洞作为数据分析对象，主要是由于近年来随着微博信息传播力和组织能力的突显，微博上面的数据更加具有开放性和及时性，同时根据最近一段时间的热点事件我们不难发现，微博已经慢慢成了热点事件快速传播的平台，正是因为这样，微博上面的数据已经吸引了各类学者的关注，本文就是基于电子科技大学微博树洞数据，完成对电子科技大学微博树洞里面人群特征，微博数据的统计和数据可视化分析。

**关键字：**数据分析，微博，可视化，统计

**ABSTRACT**

With the development of Internet technology, social network plays an increasingly important role in people's daily life, but also changed the mode of information dissemination. Now diversified platform changes from a combination of print media and television broadcast the original phase. The way people access information and the information they are interested in is becoming more and more inseparable with social networks. Accompanied by the depth of excavation by analyzing various behavior of a large number of user’s data. So as to optimize the efficiency of information transmission, saving people valuable information on their own time to extract information from the mass. And tap the potential huge commercial value. The huge success of Facebook and Twitter in foreign social networks has greatly promoted the development of domestic social networking platforms.

This paper takes the popular social networking platform Sina, micro-blog's user information and micro-blog information as the object of study, and uses Elasticsearch and Kibana to implement data analysis for micro-blog tree hole. Chose micro-blog tree as a data analysis object, mainly because in recent years with the micro-blog information dissemination force and organizational capacity of the above data highlighted, micro-blog is more open and timely, and according to the hot events recently. It is not difficult to find, micro-blog has gradually became the focus of events the rapid spread of the platform is because of this, micro-blog above data has attracted the attention of scholars of all kinds, this is the University of Electronic Science and technology of micro-blog tree based on the data of University of Electronic Science and technology of micro-blog tree hole population characteristics, data statistics and data analysis of micro-blog.

**Key Words:** Data analysis, Micro-Blog, Visualization, Statistics

目 录

第一章 毕业设计（顶岗实习）概况 1

1.1. 实习公司和岗位概况 1

1.2. 课题背景和价值 1

1.2.1 课题背景 1

1.2.2 课题价值 2

1.3. 课题整体完成情况概况 3

第二章 国内外研究状况 5

2.1 国内研究现状 5

2.2 国外研究现状 6

第三章 需求分析和系统设计 8

3.1. 需求分析 8

3.1.1 功能需求 8

3.1.2 非功能需求 10

3.1.3 数据需求 11

3.2. 系统设计 11

3.2.1 系统模块划分 11

3.2.2 技术方案制定 13

第四章 关键问题解决方案 17

4.1 数据采集技术 17

4.2 针对微博的数据分析 17

4.3 模拟登陆验证码识别 18

4.4 数据爬取和存储效率（多线程） 19

4.5 爬虫设计 21

4.6 避免反爬虫策略 24

4.7 数据分析模型 26

第五章 系统测试和数据分析结果 29

5.1 系统测试 29

5.1.1 Junit单元测试框架 29

5.1.2 系统测试用例 29

5.2. 数据分析方法 30

5.3. 微博数据结果分析 32

5.4. 后续扩展研究计划 38

第六章 职业素养与工程伦理的学习与培养 39

6.1. 个人职业素养培养 39

6.2. 软件工程对社会的影响 40

第七章 对软件工程实践以及软件工程领域发展的认识 42

7.1. 软件工程和可持续发展 42

7.2. 软件工程发展状况 42

第八章 结束语 44

参考文献 45

致 谢 46

外文资料原文 47

Sentiment Analysis of Twitter Data 47

外文资料译文 49

Twitter数据的情感分析-节选 49

# 第一章 毕业设计（顶岗实习）概况

## 1.1. 实习公司和岗位概况

**公司简介。**SAP公司成立于1972年，总部位于德国沃尔多夫市，在全球拥有6万多名员工，遍布全球130个国家，并拥有覆盖全球11,500家企业的合作伙伴网络。作为全球领先的企业管理软件解决方案提供商，SAP帮助各行业不同规模的企业实现卓越运营。从企业后台到公司决策层、从工厂仓库到商铺店面、从电脑桌面到移动终端—SAP助力用户和企业高效协作，获取商业洞见，并从竞争中脱颖而出。SAP的软件和服务能够帮助客户实现盈利性的运营，不断提升应变能力，实现可持续的增长。全球188个国家的232000家客户正在从SAP解决方案中获益, 其中包括财富500强80%的企业及85%最有价值的品牌。

**岗位职责。**参与思爱普PCM（客户渠道管理）系统的开发，负责前期接口的设计和数据库设计，负责前端和后端的实现，负责PCM集成微信和集成内部的C4C（客户云服务）系统。项目使用OData数据交换协议，后端开发框架使用Spring Boot。

## 1.2. 课题背景和价值

### 1.2.1 课题背景

随着互联网技术的飞快发展。网络在人们生活中的普及率越来越高，成为人们生活中不可以或缺的一部分，特别是微博，QQ，微信朋友圈社交网络的出现标志着互联网新格局的到来，改变了我们获取信息的方式和途径，使得我们可以更快更及时的获取信息，可以足不出户的购买我们想要获得的商品，通过各种社交平台分享和交流自己的经验和感受，实时状态和位置信息。同时越来越多的人开始享受互联网带来的便捷和乐趣。根据中国互联网信息中心（China Internet Network Information Center，简称CNNIC）于2016年1月发布的第37次《中国互联网络发展状况统计报告》显示，截至2015年12月，中国网民的规模己达到6.88亿，全年共新增网民3951万人。互联网普及率为50.3%，较2014年底提升了2.4个百分点。其中使用手机上网的网民达到6.20亿，比2014年底增加了6303万人，占总体网民人数的90.1%，比2014年増长了4.3%。由此可以看出，以手机为代表的移动设备终端已经成为了我国网民上网的主要方式。

在综合社交领域，最具代表性的朋友圈、QQ空间、微博，网民使用率分别为65.1%、80.6%、33.5%。其中QQ空间主要满足用户对个人关系链信息的需求,在产品形态和商业营销方面一直坚持变革，凭借良好的用户基础,在基于大数据的关系营销方面做了诸多有益的探索，回报显著；朋友圈则通过其优秀的个人隐私保护获得了很多青少年的青睐，并且正慢慢地获得更多较高年龄阶段的用户，称为中国目前现在占有率最高的熟人社交内容分享平台；微博则主要满足用户对兴趣信息的需求，是用户获取和分享"新闻热点"、"兴趣内容"、"专业知识"、"舆论导向"的重要平台。

微博一词译自英文单词“micro-blog”，是博客的微缩版或者变体。这一理念最早由博客先驱埃文 威廉姆斯提出并于2006年创办了世界上最早提供微博服务的网站Twitter。我国微博应用始于2007年。“无处不在的沟通”是它的理念，“What are you doing”是它的宣传口号。目前，网络时代已经来临，上网购物#查阅信息等已经代替了传统的报纸商店，随着微博的兴起，国内学术界对微博的研究日益增多，迄今为止已有许多学者从教育学，传播学，经济学等多种学科视角，运用定量分析或者定性分析以及定量与定性相结合的研究方法对微博进行分析。时隔两年，2009年国内互联网商家掀起建设微博平台热潮。2009年10月，新浪微博正式向公众开放，中国网民的微博时代到来，2010 年被称为微博元年。根据中国互联网络信息中心（CNNIC）的第31次“中国互联网络发展状况统计报告”，截至2012年12月底，我国网民规模达到5.64亿，其中微博用户规模为3.09亿，比2011年底增长了54.7％，同时手机微博用户规模达到2.02亿，占微博用户的65.6％。在这样的背景之下，微博及其相关话题成为政府、商务机构和寻常民众的热门话题，也是信息科学和社会科学共同关心的焦点之一。

高速发展的互联网产生了大量的数据，大量的数据必然会有大量有用的信息，因此，“大数据”由此成为时下最为火热的词汇之一，数据仓库，数据分析和数据挖掘以及数据安全等围绕数据商业价值的利用被各行各业的人相互追捧。大量数据的出现，正引发全球范围内的商业技术和商业变革。从海量的大数据中，我们可以快速的获得更价有意义和有价值的信息，由此衍生出了数据挖掘方面的研究。数据挖掘是一项借助工具探测海量数据中有意义的信息的一种行为，借用该过程获得的信息，我们可以很方便的支撑其它的商业和科研活动，通过充分得使用数据，挖掘数据，我们可以从中发掘出潜在问题，作出更加合理的决策。

### 1.2.2 课题价值

新浪微博（http://weibo.com）是新浪旗下网站，是中国最具影响力的社会化媒体平台，是一个基于用户关系的信息分享、传播以及获取的平台。用户通过WEB、WAP以及各种客户端，以不超过140个字的文字更新信息，实现即时分享。微博作为用户占有率排行靠前的社交平台，拥有大量的用户产生的数据，这些数据隐藏了很多有价值的信息，因此，针对微博的数据挖掘和分析工作已经引起了越来越多的人的关注和重视。微博具有良好的互动性，覆盖领域多样化，用户人数多，热点信息传播快，用户范围广的特点，因此针对微博的数据分析能够更加全面，分析微博数据所获得的信息对于实际需求也更佳有价值和意义。

通过爬取微博树洞数据，可以获取关注树洞的活跃用户的明显特征，从多个数据维度对人群进行分类，分析用户行为。例如通过分析树洞动态评论数据，可以获得关于该树洞人群的微博使用习惯数据，通过分析树洞人群的个人信息，可以获取树洞人群特征信息。除此之外，企业也可以对这些数据进行深度挖掘，获得更加具有商业价值的信息，如通过热点和用户特征进行活动和精准用户广告的投放。

目前，还没有一个系统能够直接对这样的数据进行相关的分析，本课题的系统以新浪微博为基础，通过爬虫的方式，采用多线程来快速获取微博上面的数据并且将微博数据存储在Elasticsearch中，按照微博动态内容，用户关系，用户评论数据，个人信息进行多维度的数据分析，通过用户界面能够直观的展示树洞人群特征。

同时通过这个课题，对我个人也有很大的提升，能够加深我对数据分析的学习，开扩了自己的技术眼界，从之前的开发应用程序产生数据，延伸到学习如何在数据中获取信息，利用信息产生更大的价值，让我明白数据对于提高生产力的作用。

## 1.3. 课题整体完成情况概况

**完成爬虫模块程序设计和实现。**爬虫模块支持多线程和代理池进行数据爬取，同时通过使用文件缓存的方式来记录爬虫运行时的URL爬取记录实现对爬虫的暂停，启动和终止（终止使用JMX接口实现）。能够批量爬取微博动态，评论和用户数据，能够将这些数据处理后批量发送到Elasticsearch中存储。

**完成分析数据对象建模。**本课题中的主体数据时微博动态，评论和用户，由于数据需要存储在Elasticsearch中，因此需要定义好数据存储的元数据，才能通过Elasticsearch实现对数据的分析，因此将微博动态，微博评论，微博用户建立三个数据对象模型，在持久化数据时进行数据模型的转换，将获取到的网页数据提取格式化为符合数据模型要求的数据。

**完成数据分析。**借助Kibana提供的操作Elasticsearch的用户界面和数据可视化能力，实现对获取的微博数据的分析。主要包括了用户维度信息分析，动态维度信息分析以及评论维度的信息分析，具体数据分析细节见第四章数据分析结果。

**软件工程能力培养。**通过这个项目，让我对使用软件工程开发模式管理软件开发周期有了更深层次的理解。能够使用软件工程知识来完成需求分析、项目周期管理、测试管理和持续集成。提高了我对软件工程理论的掌握和应用。

# 第二章 国内外研究状况

随着互联网技术的发展和收集数据手段的提高，微博积累的数据越来越大，形成了大数据的问题。现在常用的分析手段，如概率统计等，已经很难满足分析的需要。很多时候，人们并不知道如何来分析这些数据，因为人们看不到数据，形成了“数据盲人”。现在，可视化正作为一个新的领域蓬勃发展起来。可视化是解决这一问题的良好方法。

可视化可以粗略地被定义为通过图形的表现形式，进行信息传递、表达的过程。虽然我们今天谈到可视化一般是指利用计算机图形学和图像处理分析技术，将各种数据依据其特点转换为相应的图形图像，并提供界面实现人机交互的工作，但早在计算机发明之前，可视化就已为人类广泛应用。化技术最早运用于计算机科学中，并形成了可视化技术的一个重要分支——科学计算可视化(Visualization in Scientific Computing)。科学计算可视化能够把科学数据，包括测量获得的数值、图像或是计算中涉及、产生的数字信息变为直观的、以图形图像信息表示的、随时间和空间变化的物理现象或物理量呈现在研究者面前，使他们能够观察、模拟等。可视化的三个方向——科学可视化、信息可视化、可视分析密切相关，同时又各有特点，有其研究内涵与外延。科学可视化处理的对象包括医学、气象环境学、化学工程、生命科学、考古学、机械等领域的具有空间几何特征数据的时空现象，对测量、实验、模拟等获得的数据进行绘制，并提供交互分析手段。研究重点包括对表面、体绘制和对复杂数据中信息的选取、表达等。

## 2.1 国内研究现状

在计算机网络和通信等学科领域，微博的信息数据特征是近年来关注的主要问题，很多微博研究都基于网络信息数据测量分析展开，微博信息数据是指微博数据中心中存储的各类数据，主要包括：微博用户档案、微博用户关系、微博消息热点话题等，它是此类研究方法的基础。在这类研究中，大多以微博消息传播三大构件为研究对象，以微博消息传播和微博成员组织为主要研究内容，目的在于发现微博中的用户、消息传播、热点话题、用户关系网络等的规律。主要研究的问题如下。

1. 基于微博用户的研究，主要研究用户的行为特征及用户的影响力。

2. 基于微博用户关系的研究，主要研究用户关系网络的基本属性、关系网络生成和演进、微博人员关系挖掘、微博用户人际关系特点。

3. 基于微博内容的研究，主要研究微博消息内容特点、消息活跃时间特点、微博热点话题特点。

4. 基于微博消息传播的研究主要研究微博消息传播的特点，微博消息传播影响力。

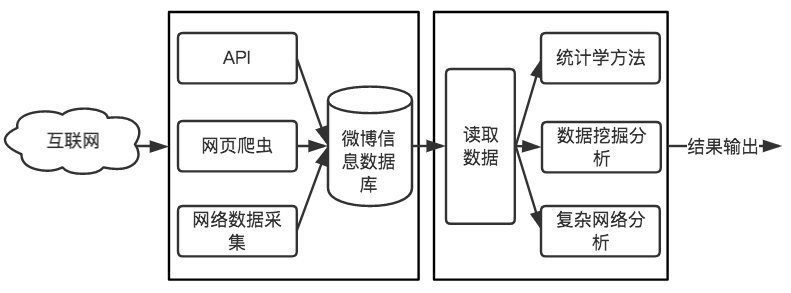
基于信息数据分析的微博研究方法，以分析微博信息数据为基础，致力于发现微博中的各种规律和特点它可以分为两个阶段: 信息数据获取和信息数据分析。在信息获取阶段，主要任务是获取大量微博信息数据，主要采用三种方法: 基于微博第三方应用程序接口(Application Programing Interface， API)编程实现微博信息爬取，利用网络爬虫在微博网页上爬取关键字信息，利用网络数据采集设备直接获取微博服务网络传输数据。在数据分析阶段，主要任务是对微博信息数据进行特征提取和分析，挖掘出微博中的关键特征，采用的主要方法包括统计学数据分析方法、复杂网络分析方法、数据分类及挖掘方法等。微博信息数据分析方法基本流程如下图2-1所示。

图2-1 微博数据分析方法

基于微博信息数据分析的研究是微博研究中非常重要的一个方向，它是开展其他微博研究的基础，能够积累大量的微博特征，对微博的理论和实践研究都至关重要。

## 2.2 国外研究现状

首先介绍Twitter的概念：Twitter是一个在线社交网络服务，允许用户发送和阅读简短的在140个字符内的消息，而这条消息称为"推文(tweet)"。

Twitter平台包含了大量的用户信息W及丰富的内容，为预测研巧提供了重要的数据基础，因此也就出现了利用微博信息进行各种问题的预测。Jurgens等人利用地理位置推理算法对Twitter用户的地理位置进行了预测并取得了较好的预测结果。Bollen等人针对股票预测问题，用每天包含正面情感词和负面情感词的推文数比值来计算情感时间序列，通过自组织模糊神经网络(Self-Organizing Fuzzy Neural Network，简称SOFNN)预测道琼斯工业平均指数(Dow Jones Industrial Average，简称DJIA)。Wolfram等人使用SVM模型，从推文文本直接提取了特征属性来进行模型训练，预测纳斯达克指数。Zhang等人分析了微博情感和DJIA变化走势之间的相关性。欧美的学者还利用Twitter数据对大选进行了预测，而且逐渐成为了研究热点，取得了较好的预测效果。

# 第三章 需求分析和系统设计

## 3.1. 需求分析

需求分析是产品研发前期的铺垫工作，也是重要的基础工作之一。需求工作中的缺陷将给项目成果带来极大风险，在推出产品时，体现在质量、功能、场景等情境下影响着用户的满意度和期望值。在软件工程的历史中，很长时间里人们一直认为需求分析是整个软件工程中最简单的一个步骤，但在过去十年中越来越多的人认识到它是整个过程中最关键的一个过程。假如在需求分析时分析者们未能正确地认识到项目的实际需求的话，那么最后的软件实际上不可能达到既定的需要，或者软件无法在规定的时间里完工。

### 3.1.1 功能需求

功能需求 （functional requirement）规定开发人员必须在产品中实现的软件功能，用户利用这些功能来完成任务，满足业务需求。功能需求有时也被称作行为需求。功能需求按照系统需要实现的主要功能可以划分为：

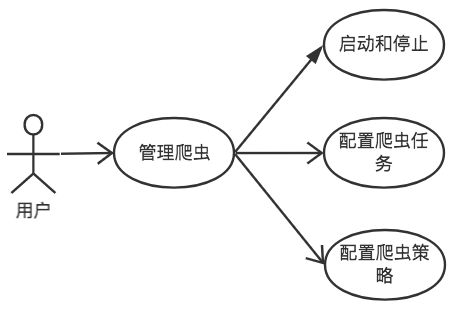
1. 微博数据爬取模块。

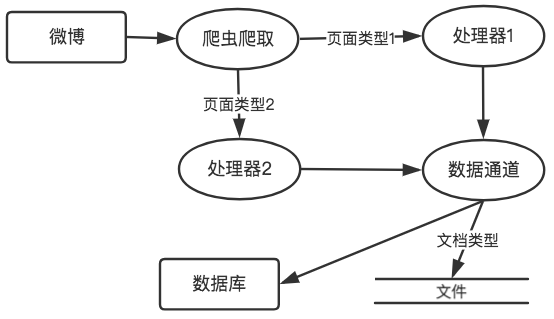
2. 数据分析功能。

3. 数据可视化模块。

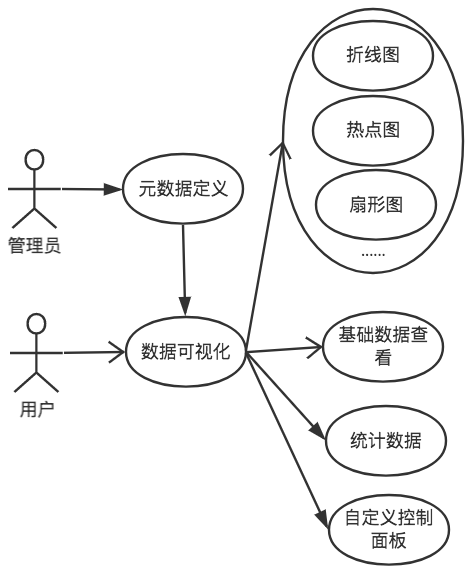
**1. 微博数据爬取模块。**微博数据爬取模块需要实现对微博动态、评论以及用户信息的获取，支持爬虫的配置文件控制，能够通过配置文件设置和改变爬虫的行为，如爬虫启动任务的数量、爬虫的请求头、爬虫主机地址、文件缓存路径和代理池等。同时该模块还应该能够实现错误处理和非法功能，能够对爬虫运行时遇到的错误进行处理并且报告，避免因为错误异常导致爬虫程序的异常退出。该模块的用例图如下图3-1所示。

**2. 数据分析功能。**微博数据分析功能主要实现将爬取的微博数据进行一定的格式化处理后持久化存储在磁盘，以实现微博数据分析功能。该功能可以使用传统的数据库实现数据的存储，也可以使用基于文档的存储引擎进行数据的存储。数据分析功能需要将爬虫获取的数据进行格式化，并且能够响应用户界面的组合查询，将查询结果返回给用户界面渲染。数据分析功能的数据流图如下图3-2所示。

图3-1 爬虫模块用户用例图

图3-2 微博数据流图

**3. 数据可视化模块。**数据可视化模块主要是对分析后的数据进行可视化，包括常用的折线图、柱状图、扇形图和热点图等，从而能够一目了然的从数据中获取有用的信息，该模块应该实现用户可以自定义数据可视化的方式，并且支持用户扩展对数据进行进一步的查询组合。可视化可以通过APP和网页两种方式提供，但是通过网页提供，应该能够支持通过媒体查询实现页面的自适应。该模块的用例图如下图3-3所示。

图3-3 数据可视化模块用例图

### 3.1.2 非功能需求

非功能性需求是指依一些条件判断系统运作情形或其特性，而不是针对系统特定行为的需求。包括安全性、可靠性、互操作性、健壮性、易使用性、可维护性、可移植性、可重用性、可扩充性。

本课题设计程序的非功能需求主要包括：

用户界面美观，具有针对性。所有页面应该支持主流的浏览器（IE9.0+，Chrome，Firefox等）下正常显示，而且不同浏览器看到的页面外观应该一致。

在操作上要易于使用。使用合理的存储方式来存储数据，保证数据的独立性，也保证了系统的可移植性。

数据存储量。使用合理的存储硬件，能够易于存储爬虫爬去的零散数据，确保不会因为硬件原因导致数据存储出错，进而影响数据分析的完成。

系统可靠性。在实现一定的吞吐量的基础上，需要满足系统运行期间不会出现严重错误导致异常退出，从而导致系统不可用、数据丢失等严重故障。

可扩展性。可扩展性主要指微博爬虫的配置应该足够灵活，且内部功能应该能够支持扩展和重用，如果有其他数据分析的需求，核心处理代码应该能够重用。另外一方面，系统在实现时应该考虑到日后若信息量较大，则系统可相应增加服务器实现扩展。

### 3.1.3 数据需求

功能需求和非功能需求都集中在系统需要做什么的层面上，对于该课题任务的主体数据却没有进行更多的定义，因此这一小节主要讨论系统的数据需求。

数据需求的基本定义：数据需求分析是从对数据进行组织与存储的角度，从用户视图出发，分析与辨别应用领域所管理的各类数据项（Data Items）和数据结构，形成数据字典的主要内容。

本课题中的三个主体数据对象为微博动态、微博评论、微博用户。数据分析对于这三个主题的数据对象应该包含足够的原始数据量。有效数据的比例应该大于90%，数据格式定义应该固定，类型使用合理。

## 3.2. 系统设计

### 3.2.1 系统模块划分

大型软件项目开发中，模块的划分非常的重要。一般来说，模块在功能上尽可能单一和明确化，最好做到模块与目标的一一对应。模块间的联系应尽可能少，对于必须的联系都应该有明确的说明，模块应足够小，以方便单个模块的调试。

对初步的架构轮廓作第二个方面的审视，是考虑模块化的设计问题。也就是从架构的组成单元来说，定义清楚子系统以后，下一步就是定义模块。

**1．模块化设计基本概念和准则**

如何合理的进行模块设计呢？这里的关键是要保证模块的独立性。模块：模块是数据说明、可执行语句等程序对象的集合，是单独命名的并且可以通过名 字来访问，例如过程、函数、子程序、宏等。模块化：软件被划分成独立命名和可独立访问的被称作模块的构件，每个模块完成一个 子功能，它们集成到一起可以满足问题需求。利用模块化解决方案的注意事项：一般来说，倾向于每个用户描述定义一个模块。我们应该努力使每个模块的大小差别在一个数量级之内。如果发现某个模块规模太大，就需要实现模块切割，然后针对这种切割的结果，反过来修改需求分析的时候用户描述（或者用例）的表达方式。这就是由设计引发的需求变更。

模块的大小一般以一个开发团队在一次迭代时间内能完成为好。模块切割方法与开发成本有关。我们可以这样来思考模块化对软件工作量和成本的影响。实际的情况见下图，随着模块数量的增加，开发成本减低，但是系统集成的成本增加，所以最小成本的区域在一个合适的区间。也就是说，模块并不是越多越好，只有模块数 量适当的时候，总体成本才可能下降。

改进软件结构，提高模块独立性。模块规模应该适中。大模块分解不充分；小模块使用开销大，接口复杂。尽量减少高扇出结构的数目，随着深度的增加争取更多的扇入。扇出过大意味着模块过分复杂，需要控制和协调过多的下级模块。一般来说，顶层扇出高，中间扇出少，低层高扇入。模块的作用范围保持在该模块的控制范围内。

模块的作用范围是指该模块中一个判断所影响的所有其它模块；模块的控制范围指该模块本身以及所有直接或间接从属于它的模块。力争降低模块接口的复杂程度模块接口的复杂性是引起软件错误的一个主要原因。接口设计应该使得信息传递简单并且与模块的功能一致。设计单入口单出口的模块，以避免内容耦合，易于理解和维护。模块的功能应该可以预测。相同的输入应该有相同的输出，否则难以理解、测试和维护。

**2．实现模块化的手段**

抽象：抽出事物的本质特性而暂时不考虑它们的细节。

信息隐蔽：应该这样设计和确定模块，使得一个模块内包含的信息（过程和数据）对于不需要这些信息的模块来说，是不可访问的。

模块独立性问题：模块独立是指开发具有独立功能而且和其它模块之间没有过多的相互作用的模块。

**3．本系统模块设计**

本系统主要包括四个模块：爬虫模块，数据处理模块，数据分析模块以及数据可视化模块。

**爬虫模块。**爬虫模块采用python或者java实现，需要能够实现爬取微博的动态数据和评论数据，并且将爬取后的数据进行简单的处理。同时为了防止微博反爬虫的限制，需要在模拟请求头中设置字段，保证发出去的请求和浏览器发出去的请求大体一致，设置代理连接池，循环切换代理线路或者在一定的次数内使用某条代理线路然后切换，避免同一IP地址频繁访问，被微博拦截。

**数据处理模块。**数据处理模块主要负责将爬取到的数据进行初步的分析，将不同的数据对象存储到Elasticsearch中。

**数据存储模块。**数据分析模块主要是由Elasticsearch完成，ElasticSearch是一个基于Lucene的搜索服务器。它提供了一个分布式多用户能力的全文搜索引擎，基于RESTful Web接口。通过Elasticsearch提供的DSL查询语言来对Elasticsearch中的数据进行统计搜索查询。

**数据可视化模块。**数据可视化模块使用Kibana，Kibana是一个开源和免费的工具，可以汇总、分析和搜索重要数据日志并提供友好的web界面。

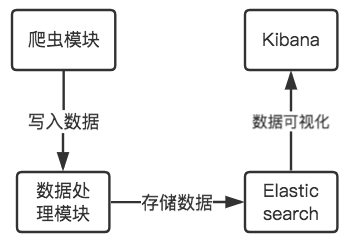
本系统的模块设计如下图3-4所示

图3-4 系统模块划分

### 3.2.2 技术方案制定

在确定了需求后，需要在需求分析的基础上制定大致的技术方案。

在选择技术方案时，技术先进性不再是唯一的指标，也要比较不同技术的对于项目的适应性，在同样或差别不大的情况下，应该优先选择更佳适宜本项目的技术方案，不能因为一昧的追求新技术，避免技术方案选得过于激进，而忽视了对于本项目实际需求的理解。

**1. 依赖管理和打包工具-maven。**

项目的依赖管理使用maven来实现，maven是一个软件（特别是Java软件）项目管理及自动构建工具，由Apache软件基金会所提供。基于项目对象模型（缩写：POM）概念，Maven利用一个中央信息片断能管理一个项目的构建、报告和文档等步骤。

Maven也可被用于构建和管理各种项目，例如C#，Ruby，Scala和其他语言编写的项目。Maven曾是Jakarta项目的子项目，现为由Apache软件基金会主持的独立Apache项目。

项目使用maven来管理所以来的其他jar包，同时使用maven来生成合理的目录结构，方便后期的开发，迭代和维护，在部署时，也可以通过maven的自动构建完成项目的打包，将项目打包成可执行文件后进行部署。除此之外，还可以使用maven的profile管理功能，为线上生产环境和本地开发环境设置不同的profile，在执行打包时，选择对应的profile即可完成自动区分并且生成适用于不同环境的可执行jar包文件。

**2. 集成开发工具-IntelliJ IDEA**

项目使用IntelliJ IDEA来进行开发。 IDEA 全称IntelliJ IDEA，是java语言开发的集成环境，IntelliJ在业界被公认为最好的java开发工具之一，尤其在智能代码助手、代码自动提示、重构、J2EE支持、Ant、JUnit、CVS整合、代码审查、 创新的GUI设计等方面的功能可以说是超常的。

**3. 代码管理-git**

为了管理代码的历史版本，方便对代码进行回滚，版本记录等功能，本项目使用git作为代码管理工具。

git是用于Linux内核开发的版本控制工具。与CVS、Subversion一类的集中式版本控制工具不同，它采用了分布式版本库的作法，不需要服务器端软件，就可以运作版本控制，使得源代码的发布和交流极其方便。git的速度很快，这对于诸如Linux内核这样的大项目来说自然很重要。git最为出色的是它的合并追踪（merge tracing）能力。

**4. 爬虫框架-WebMagic**

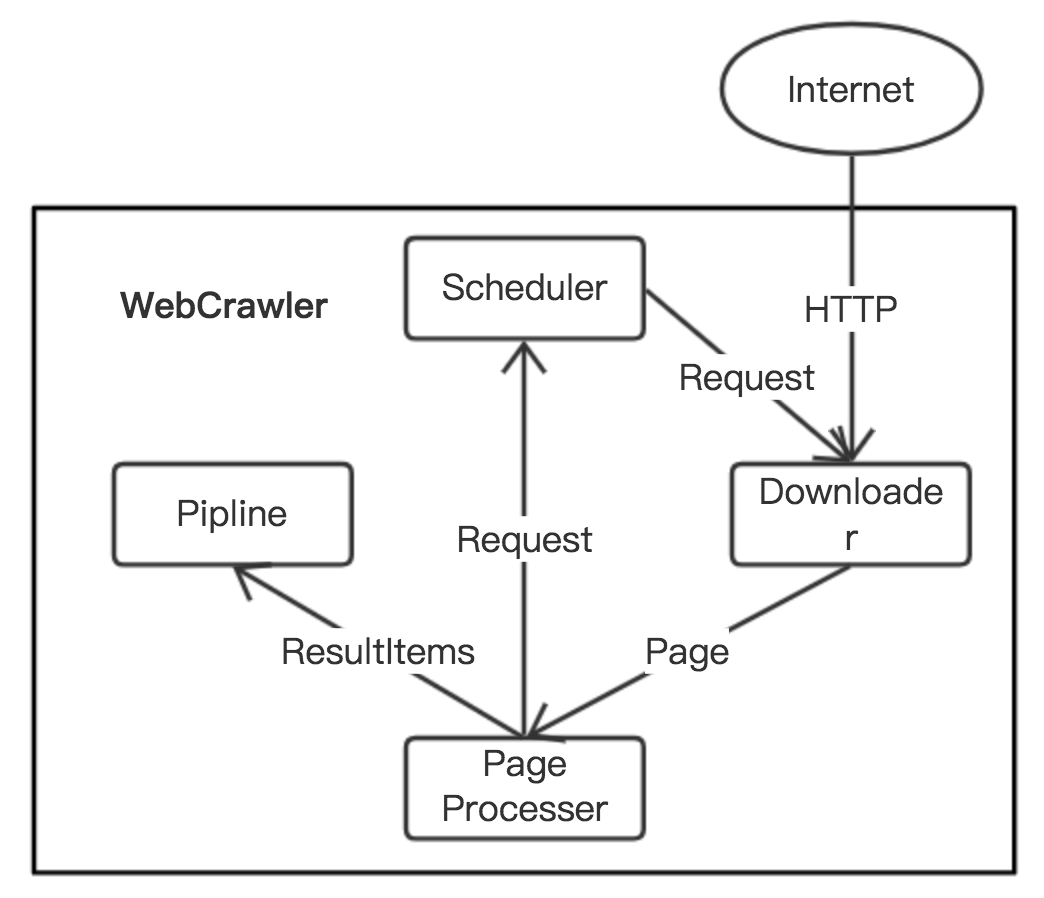
项目采用WebMagic爬虫框架，WebMagic的设计参考了业界最优秀的爬虫Scrapy，而实现则应用了HttpClient、Jsoup等Java世界最成熟的工具，WebMagic使用Java原生的开发方式，只不过提供了一些模块化的约束，封装一些繁琐的操作，并且提供了一些便捷的功能。WebMagic总体架构图如图3-5所示：

图3-5 WebMagic总体架构图

WebMagic包含四个组件：Downloader，Page Processor，Scheduler，Pipeline，四个组件实现的功能如下：

1. Downloader。Downloader负责从互联网上下载页面，以便后续处理。WebMagic默认使用了Apache HttpClient作为下载工具。

2. Page Processor。Page Processor负责解析页面，抽取有用信息，以及发现新的链接。WebMagic使用Jsoup作为HTML解析工具，并基于其开发了解析XPath的工具Xsoup。在这四个组件中，Page Processor对于每个站点每个页面都不一样，是需要开发者定制的部分。

3. Scheduler。Scheduler负责管理待抓取的URL，以及一些去重的工作。WebMagic默认提供了JDK的内存队列来管理URL，并用集合来进行去重。也支持使用Redis进行分布式管理。

4. Pipeline。Pipeline负责抽取结果的处理，包括计算、持久化到文件、数据库等。WebMagic默认提供了“输出到控制台”和“保存到文件”两种结果处理方案。Pipeline定义了结果保存的方式，如果需要保存到指定数据库，则需要编写对应的Pipeline，对于一类需求一般只需编写一个Pipeline。

**5. 数据存储和可视化实现方案-Elasticsearch，Kibana**

数据可视化是将纯文本的数据信息通过图像的形式更加形象地表现出来，从而数据阅读者能够一目了然地获取到关键信息。

前端实现数据可视化主要是通过一些数据可视化插件来实现，如HighCharts，Echarts等，这些插件通过内置数据可视化图表，使得我们可以快速的通过定义后端数据接口来实现数据可视化。

但是本项目的研究重点不是如何实现数据可视化，而是从爬取的微博数据中获得有价值的信息，因此我们采用了Elasticsearch和Kibana来快速实现数据可视化。Elasticsearch提供数据存储，索引和查询服务，Kibana消费Elasticsearch接口来实现数据的可视化。

所以数据最后是存储在Elasticsearch中进行索引和查询的，因此爬虫模块在获得数据后需要将数据通过Elasticsearch的API将数据持久化到Elasticsearch中，因此需要确定良好的满足Elasticsearch要求的数据结构，将爬虫提取到的数据进行一定的处理后再存储在Elasticsearch中，由于之前没有使用过Elasticsearch，因此需要阅读研究Elasticsearch的文档，从而完成爬虫数据的存储，这一步对于后期的数据分析有着很重要的影响，只有确定了合理的存储方式和粒度后，后期的数据分析才能展现多维度的信息，我们也才能够从中提取到更多有价值的信息。

为什么使用Elasticsearch？

Elasticsearch是一个基于Apache Lucene(TM)的开源搜索引擎。无论在开源还是专有领域，Lucene可以被认为是迄今为止最先进、性能最好的、功能最全的搜索引擎库。

但是，Lucene只是一个库。想要使用它，必须使用Java来作为开发语言并将其直接集成到你的应用中，更糟糕的是，Lucene非常复杂，你需要深入了解检索的相关知识来理解它是如何工作的。

Elasticsearch也使用Java开发并使用Lucene作为其核心来实现所有索引和搜索的功能，但是它的目的是通过简单的RESTful API来隐藏Lucene的复杂性，从而让全文搜索变得简单。

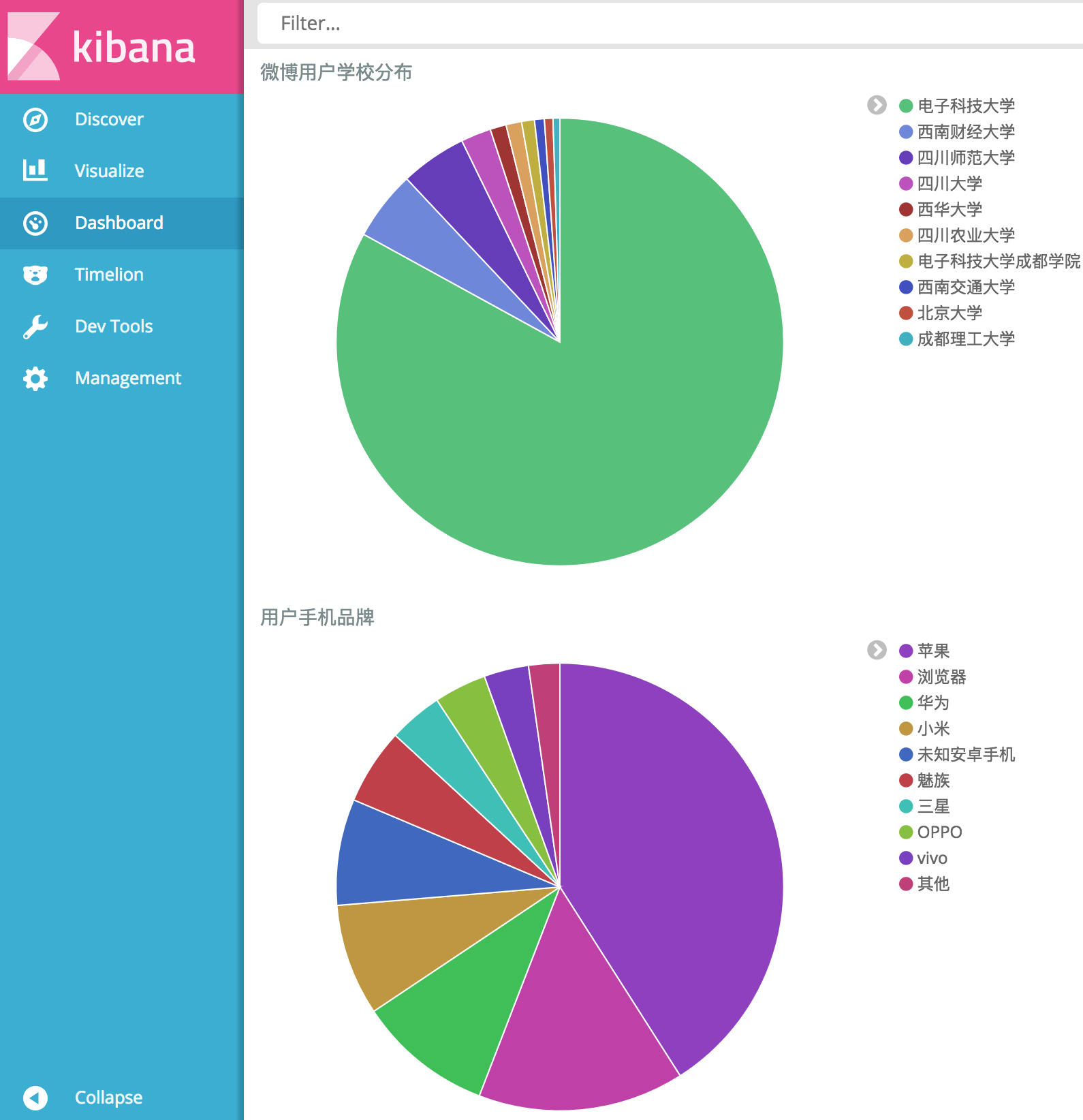
使用Kibana可以方便的提供组合查询和数据可视化功能，可以大大提高我们数据分析工作的效率，使得我们的关注点在数据分析本身，而不是过多的去关注如何实现数据可视化。Kibana数据可视化界面如下图3-6所示。

图3-6 Kibana数据可视化界面

# 第四章 关键问题解决方案

## 4.1 数据采集技术

目前的数据采集技术主要有以下四种。

**1. 基于网站的信息抓取。**这种方法是从一些基础URL出发，采用深度优先或者广度优先策略进行爬取，逐步扩展到整个网站。这种方法的优点是信息全、覆盖面广，主要被各大搜索引擎所使用，其缺点是爬行耗时长，不利于最新信息的搜集，此外由于所需的硬件资源多，只有少数几个服务商能够采用这种技术。

**2. 基于主题的信息爬取。**这种爬虫的特点是先进行网络内容的判断，如果与想要爬取的主题一致，再进行爬取。由于只对某一个主题进行爬取，所以此种爬虫的优点是采集的页面针对性很强，效率也较高，比较适用于专门对某一个问题进行研宄的研究人员。如果想要先采集数据，再进行数据分析，则此种爬虫不适用。

**3. 增量式信息爬取。**此种爬虫是针对第一种网络的对新信息反应慢的缺点而提出的。它只对有变化的网页进行再爬取，而不对没有变化网页进行爬取。此种爬虫的效率高，在信息量大的今天有很强的实际意义，它的难点在于如何判断网页是否有变化，所以它的爬行效果完全取决于所使用的判断算法的有效性。

**4. 分布式网络爬虫。**由于目前网络上的信息量巨大，如果只使用一台计算机进行采集，耗时巨大，信息更新也慢，所以研宄人员提出了使用很多台计算机同时对网络进行信息采集，这样一来可以大大提高爬虫的速度和效率，但是如何进行爬行任务的分工以及多台计算机如何协调配合，使每台的工作压力相对均衡是此种爬虫的设计难点。

本课题使用基于主题的信息爬取，采集数据时获取网页中的链接，再通过正则匹配获得包含目标数据的页面地址，并且将地址加入爬虫待爬取地址池，最后爬虫从筛选出来的合法网址中提取目标数据，完成数据采集。

## 4.2 针对微博的数据分析

目前已经有很多针对微博而进行的数据分析工作，利用微博的数据，通过一定的数据分析手段，从中获得有价值的信息，常见的几种数据分析目的主要有以下三点：

**1. 微博短文本挖掘。**由于微博的长度较短，所以可以把微博上的内容看成是不同短文本的集合，这个集合有一些显而易见的特点:文本口语化，存在大量错别字、谐音字、符号等;信息碎片化，文本的特征不明显，意思表达不清楚;信息量巨大，又由于转发量大，原创数量小，微博内容的重复性很高。为弥补短文本特征值的不足，S.Bharath将用户所发的微博按所涉领域进行划分，然后对不同微博进行分类，这样提高了短文本的特征值。而针对微博内容的重复性，王永恒提出了用于聚合大量短文本的方法，通过聚合，可以使得重复性的文本大大减少，提高了剩余文本间的区分度

**2. 情感倾向性分析。**人们在所发的微博中，通常都带有自己的情感倾向。这些情感倾向通过使用语气词、具有判断性的词汇表达出来，情感倾向对于研究诸如用户满意度、大众对社会热点的态度等尤为重要。分析人们的情感倾向，最常用的方法就是建立情感词库，然后进行词汇的匹配和表情、语气等的标记，以此来发现人们的倾向另外，也出现了一些情感分类的算法，如最大信息熵、向量机、贝叶斯算法等，在使用此类算法时，为了提高正确率，除了进行情感的分类外，还需要结合上下文进行判断。

**3. 话题趋势检测。**由于微博具有快速更新的特点，一个热点话题可以在极短的时间内广泛传播，造成巨大的影响。所以快速的找出热门话题，判断话题传播的方向、发展的趋势等，就能更好的掌握舆论的走势，为下一步的行动争取时间。最常见的话题趋势检测就是进行词频的检测，通过检测某个词出现的频率，就能判断某个话题的热门程度。除此之外，可以结合PageRank算法，将转发量、作者的重要性、词频等作为加权因素，就可以计算出此话题的重要性，并判断其未来的走势。

本课题主要通过爬虫获取微博动态，评论和用户数据，分析数据的特点，从中获取有价值的数据。

## 4.3 模拟登陆验证码识别

验证码识别方案主要有如下三种：

**1. 通过图像识别技术来识别验证码。**该方法的主要的实现方式有二值化、去干扰线和分割字符。这种方式准确率不高，且实现难度较高，而且验证码识别不是本课题的研究重点，因此不采用这种方式。

**2. 通过第三方服务识别验证码。**由于验证码识别在市场上有一定的需求，因此有很多第三方的验证码识别解决方案，第三方识别验证码主要是通过机器学习识别和人工识别。前者的准确率受到服务提供商的技术水平的影响，后者准确率很高。但是通过第三方服务来进行验证码识别使得我们的程序依赖于第三方服务，且要收费，不是很合理的选择，因此不采用这种方式。

**3. 本地识别。**本地识别主要是在程序判断需要输入验证码的情况后，将验证码图片下载到本地，然后提示用户读取本地文件的验证码上面的信息，然后输入到程序使程序继续执行，通过这种交互的方式来实现验证码的识别。由于微博的验证码识别不是一个频发事件，且程序通过设置代理和间隔休眠会进一步减少需要输入验证码的情况，因此本课题采用这种方式来实现验证码识别。

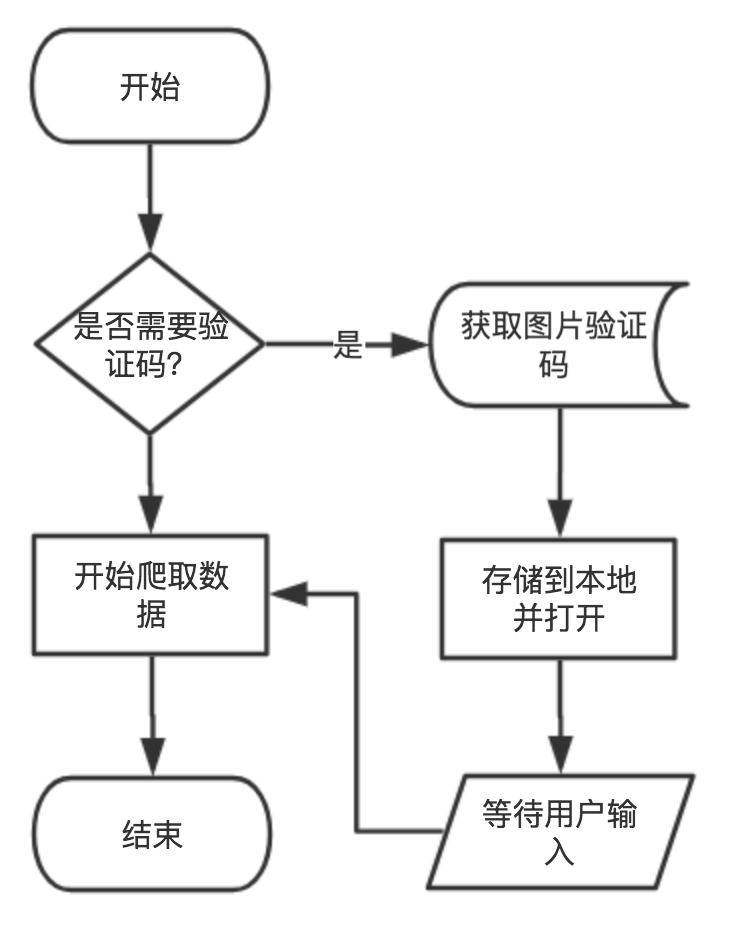
识别步骤：当获取到需要输入验证码的返回后，爬虫从微博服务器请求验证码图片并且存储在本地，并且暂停执行，提示并等待用户输入验证码图片中的验证码信息，将验证码信息发送至微博服务器验证，识别成功则唤起爬虫线程继续执行。验证码识别步骤如下图4-1所示：

图4-1 爬虫处理验证码流程图

## 4.4 数据爬取和存储效率（多线程）

通过爬虫爬取海量的网络数据时，一方面我们需要确保爬虫不被网站服务器封掉，一方面也要提高爬虫的采集效率。

避免爬虫不被封掉，一般我们通过大量的代理IP构成代理池，通过代理来访问采集的网站，而对于如何提高爬虫的采集效率，就有很多种方法，比如：使用多进程、多线程、分布式、协程等等。本课题使用多线程来实现提高爬虫效率。

多线程基本概念：是指从软件或者硬件上实现多个线程并发执行的技术。具有多线程能力的计算机因有硬件支持而能够在同一时间执行多于一个线程，进而提升整体处理性能。在一个程序中，这些独立运行的程序片段叫作“线程”（Thread），利用它编程的概念就叫作“多线程处理（Multithreading）”。具有多线程能力的计算机因有硬件支持而能够在同一时间执行多于一个线程，进而提升整体处理性能。

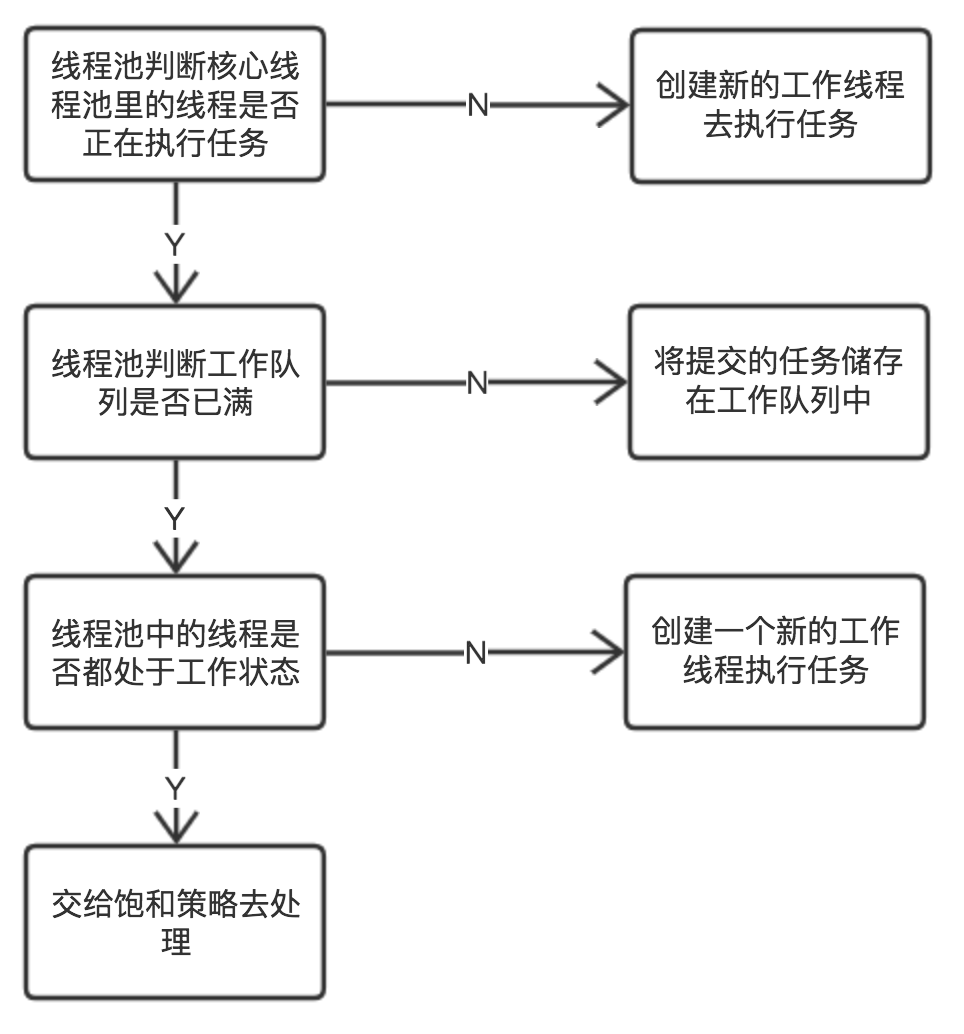
在java中多线程的实现主要是通过集成Thread类或者实现Runable接口来实现。为了使程序设计更加合理，使用Executors.newFixedThreadPool(int threadNum)来初始化一个固定长度的线程池，通过这种方式可以控制线程的最大并发数，超出的线程会在队列中等待执行。Java中线程池执行流程图如下图4-2所示：

图4-2 java线程池执行流程

通过多线程配合代理池的方式可以显著的提高爬取数据的效率，能够在短时间内采集我们所需要的数据。同样的，为了提高数据的存储效率，我们也同样使用多线程来处理已经爬取的数据，提高数据写入Elasticsearch的效率。Elasticsearch提供的BulkProcessor提供了多线程并发情况下的线程同步，使得我们可以使用多线程向其写入数据，可以设置BulkProcessor的批量提交数量，提交数据大小，提交时间间隔，只要当条件满足以上三点的任意一点，就会触发批量提交数据到Elasticsearch。

## 4.5 爬虫设计

**1. 关键部分实现**

项目使用WebMagic框架来实现微博爬虫功能，使用Elasticsearch提供的Java API来实现将格式化后的数据存储在Elasticsearch中。

不同的爬虫页面采用不同的PageProcessor来处理，处理器使用策略器模式设计，策略模式作为一种软件设计模式，指对象有某个行为，但是在不同的场景中，该行为有不同的实现算法。通过策略选择器WeiboProcessorContext来根据页面类型选择对应的Processor处理页面，UML类图设计如下图4-3所示：

图4-3 页面处理器设计

由于页面处理器没有自己的成员变量，且不同的页面可以复用相同的页面处理器，所以页面处理器属于无状态对象，是线程安全的，因此，WeiboProcessorContext策略选择器使用了一个HashMap来缓存已经实例化的Processor，避免为页面单独实例化页面处理器会由于页面数量的增多造成系统开销过大，代码实现如代码段4-1所示。

代码4-1 页面处理器代码段

|  |
| --- |
| /\*\*  \* Created by bradykang on 4/19/2017.  \* 微博页面处理策略选择类  \*/  public class WeiboProcessorContext {  private Map<PageType,WeiboProcessor> weiboProcessorMap;  public WeiboProcessorContext(){  this.weiboProcessorMap=new HashMap<PageType, WeiboProcessor>();  }  public void process(PageType pageType,Page page){  WeiboProcessor weiboProcessor=this.weiboProcessorMap.get(pageType);  if (weiboProcessor==null){  weiboProcessor=this.getWeiboProcessorByPageType(pageType);  weiboProcessorMap.put(pageType,weiboProcessor);  }  weiboProcessor.process(page);  }  } |

**2. 爬虫工作流程**

爬虫模块主要有四个参与者对象组成：

Scheduler：爬虫模块中负责调度其他对象的调度器，通过调度器来决定什么时候下载网页，什么时候调用Processor，以及将Processor处理后的数据写入到Pipline中。

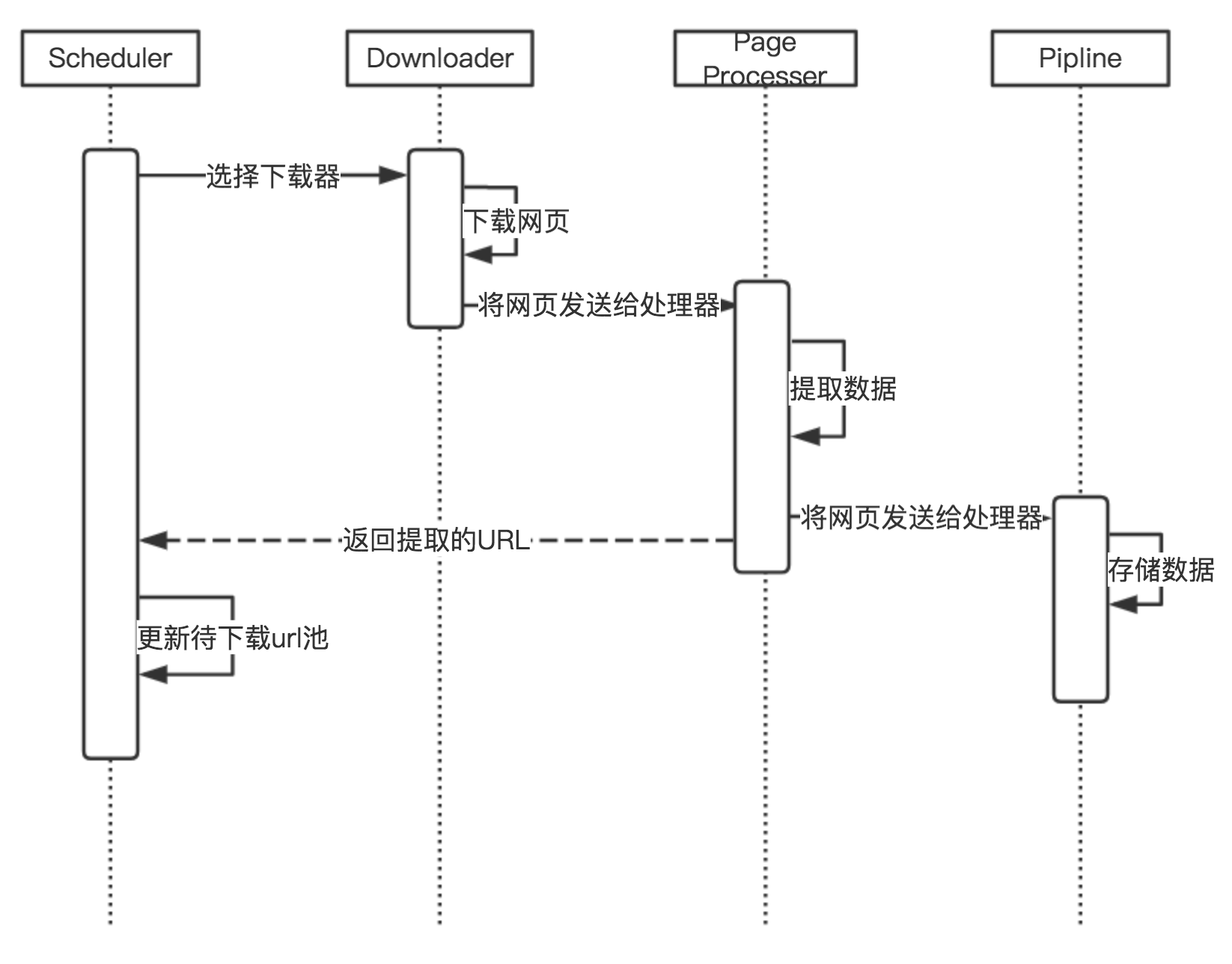
Downloader：被调度器调用完成下载网页的功能，可以为每个请求设置不同的代理以及随机请求头字段。

Page Processor：页面处理器，负责将下载器下载的网页进行处理，提取出需要的数据。

Pipline：数据管道，将页面处理器处理后的数据存储下来或者直接输出到控制台，支持同时添加多个Pipline以实现将数据输出到多个目的地。将数据格式化成json后存储到文件的Pipline实现如下代码段4-2所示。

代码4-2 JsonFilePipline实现

|  |
| --- |
| public class JsonFileExPipeline extends  FilePersistentBase implements Pipeline {  private Logger logger = LoggerFactory.getLogger(this.getClass());  public JsonFileExPipeline() {  this.setPath("/data/webmagic");  }  public JsonFileExPipeline(String path) {  this.setPath(path);  }    public void process(ResultItems resultItems, Task task) {  String path = this.path + PATH\_SEPERATOR + task.getUUID() + PATH\_SEPERATOR;  try {  PageType pageType=resultItems.get("type");  PrintWriter e = new PrintWriter(new FileWriter(this.getFile(path + pageType+ "-" + DigestUtils.md5Hex(resultItems.getRequest().getUrl()) + ".json")));  e.write(JSON.toJSONString(resultItems.getAll()));  e.close();  }  catch (IOException var5) {  this.logger.warn("write file error", var5);  }  }  } |

图4-4 爬虫爬取网页流程图

## 4.6 避免反爬虫策略

**避免IP限制。**为了避免新浪微博反爬虫的限制，一方面系统通过设置合理的爬取策略。经过测试，新浪微博会对五分钟内，持续超过3秒一次的请求进行屏蔽，为了避免这种情况，爬虫模块通过代理池来请求新浪微博页面数据，从而避免微博反爬虫限制通过IP对爬虫进行封锁。

**避免通过账户限制。**新浪微博会限制同一账户短时间内多次进行相同操作，为了避免爬虫在爬取数据时新浪微博封锁爬虫账号，因此系统维护了一个可用的账号池来实现当新浪微博屏蔽当前帐号时自动切换可用的账号以继续完成爬取任务。

**避免通过请求头限制。**很多网站都会对Headers的User-Agent进行检测，还有一部分网站会对Referer进行检测（一些资源网站的防盗链就是检测Referer）。如果遇到了这类反爬虫机制，可以直接在爬虫中添加随机Headers。系统重新实现了Downloader，使得其支持为每个请求定义单独的请求头，并且通过维护一个随机请求头信息数组来实现每次发起请求时随机选取请求头数据组装请求包。代码实现如下代码段4-3所示。

代码4-3 爬虫使用随机请求头代码段

/\*\*

\* Created by bradykang on 4/21/2017.

\* 扩展{@link HttpClientDownloader}，增加支持对每个请求设置请求头，<br/>

\* {@link Request}中设置的请求头优先级大于{@link Site}中设置的请求头

\*/

public class HttpClientExDownloader extends HttpClientDownloader {

protected HttpUriRequest getHttpUriRequest(Request request, Site site, Map<String, String> headers,HttpHost proxy) {

RequestBuilder requestBuilder = selectRequestMethod(request).setUri(request.getUrl());

Map<String ,String> allHeaders=new HashMap<String, String>();

if (headers!=null){

allHeaders.putAll(headers);

}

Object object=request.getExtra("headers");

if (object!=null){

Map map= (Map) object;

for (Object key:map.keySet()){

allHeaders.put(String.valueOf(key),String.valueOf(map.get(key)));

}

}

for (Map.Entry<String, String> headerEntry : allHeaders.entrySet()) {

requestBuilder.addHeader(headerEntry.getKey(), headerEntry.getValue());

}

requestBuilder.addHeader("User-Agent",this.getRandomUserAgent());

RequestConfig.Builder requestConfigBuilder = RequestConfig.custom()

.setConnectionRequestTimeout(site.getTimeOut())

.setSocketTimeout(site.getTimeOut())

.setConnectTimeout(site.getTimeOut())

.setCookieSpec(CookieSpecs.BEST\_MATCH);

if (proxy !=null) {

requestConfigBuilder.setProxy(proxy);

request.putExtra(Request.PROXY, proxy);

}

requestBuilder.setConfig(requestConfigBuilder.build());

return requestBuilder.build();

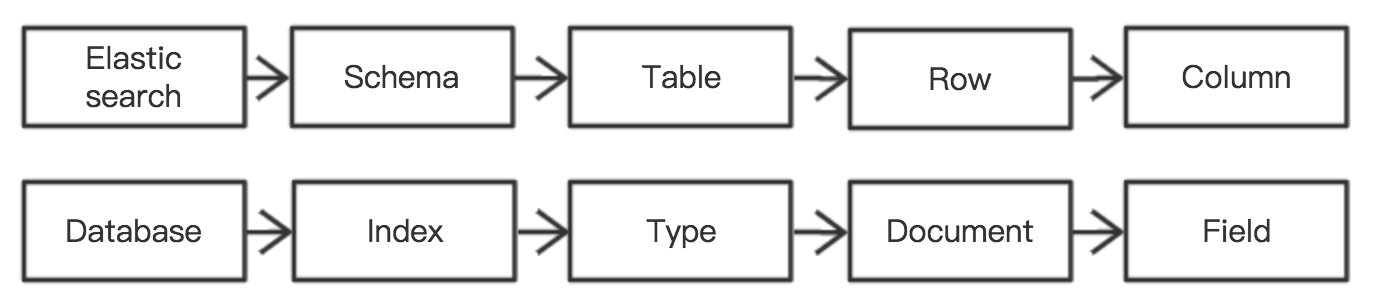
}

}

## 4.7 数据分析模型

在Elasticsearch中，数据是以文档的方式存在，Elasticsearch为每份文档建立索引方便可以根据关键字快速查找包含该关键字的文档，每份文档需要定义用来描述该文档的mapping元数据，该mapping里面定义了文档的字段属性，搜索行为，索引行为以及格式化信息等，通过这种方式，可以充分的利用Elasticsearch提供的文件搜索功能来对数据进行各种条件聚合查询，同时Elasticsearch还为每个文档记录了一个版本号用来标识该文档在Elasticsearch的改动，我们可以通过这个特性完成对Elasticsearch中文档的更新。

在Elasticsearch中，包含两种数据结构，索引（Index），文档类型（Type）。Elasticsearch数据结构和数据库的数据结构对照如下图4-5所示。

图4-5 Elasticsearch和数据库存储结构对照表

Elasticsearch是面向文档(document oriented)的，这意味着它可以存储整个对象或文档(document)。然而它不仅仅是存储，还会索引(index)每个文档的内容使之可以被搜索。在Elasticsearch中，可以对文档（而非成行成列的数据）进行索引、搜索、排序、过滤。这种理解数据的方式与以往完全不同，这也是Elasticsearch能够执行复杂的全文搜索的原因之一。

为了提高Elasticsearch索引文档的效率，需要为存储的不同类型的文档设置文档映射，通过映射来告诉Elasticsearch如何去索引文档，Elasticsearch映射定义了字段类型，每个字段的数据类型，以及字段被Elasticsearch处理的方式。映射还用于设置关联到类型上的元数据。

**1. 微博用户（User）。**微博用户在Elasticsearch中的数据存储模型如下表哦4-1所示。

表4-1 微博用户数据模型

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否分词 | 字段含义 | 是否为空 | 默认值 |
| id | Integer | 否 | 用户id | 否 |  |
| nickname | String | 是 | 用户昵称 | 是 |  |
| tag | String | 是 | 用户标签 | 是 |  |
| sex | Integer | 否 | 性别 | 是 |  |
| location | String | 是 | 地址 | 是 |  |
| school | String | 是 | 学校 | 是 |  |
| email | String | 是 | 邮箱 | 是 |  |
| level | Integer | 否 | 等级 | 是 | 0 |
| joinTime | Date | 否 | 加入时间 | 是 |  |
| intro | String | 是 | 个人简介 | 是 |  |
| company | String | 是 | 公司 | 是 |  |
| credit | String | 是 | 信用状况 | 是 |  |
| blog | String | 是 | 博客 | 是 |  |

**2. 微博评论（Comment）。**微博评论在Elasticsearch中的数据存储模型如下表4-2所示。

表4-2 微博评论数据模型

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否分词 | 字段含义 | 是否为空 | 默认值 |
| id | Integer | 否 | 用户id | 否 |  |
| create\_time | Date | 否 | 创建时间 | 是 |  |
| like\_counts | Integer | 否 | 赞的数量 | 是 | 0 |
| liked | Boolean | 否 | 是否已赞 | 是 | False |
| source | String | 是 | 来源设备 | 是 |  |
| text | String | 是 | 评论内容 | 是 |  |
| time | Date | 否 | 评论时间 | 是 |  |
| formatSource | String | 是 | 来源设备 | 是 |  |
| User | Object | 否 | 用户 | 否 |  |

**3. 微博动态（Feed）****。**微博动态在Elasticsearch中的数据存储模型如下表4-3所示。

表4-3 微博动态数据模型

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否分词 | 字段含义 | 是否为空 | 默认值 |
| attitudes\_count | Integer | 否 | 赞同数量 | 是 | 0 |
| bid | Integer | 否 | Id | 否 |  |
| idstr | String | 是 | Id | 否 |  |
| comment\_count | Integer | 否 | 评论数量 | 是 | 0 |
| create\_at | Date | 否 | 创建时间 | 是 |  |
| favorited | Boolean | 否 | 是否已赞 | 是 | False |
| isLongText | Boolean | 否 | 是否长文 | 是 |  |
| mblogtype | Integer | 否 | 动态类型 | 是 |  |
| mid | String | 是 | Id | 否 |  |
| reposts\_count | Integer | 否 | 重试次数 | 是 | 0 |
| source | String | 是 | 来源设备 | 是 |  |
| text | String | 是 | 内容 | 是 |  |
| topic | String | 是 | 主题 | 是 |  |
| create\_time | Date | 否 | 创建时间 | 是 |  |

# 第五章 系统测试和数据分析结果

## 5.1 系统测试

### 5.1.1 Junit单元测试框架

系统采用Junit单元测试框架来完成对爬虫和Elasticsearch访问功能的测试。JUnit是一个回归测试框架，被开发者用于实施对应用程序的单元测试，加快程序编制速度，同时提高编码的质量。JUnit 测试框架能够轻松完成以下任意两种结合：

1. Eclipse 集成开发环境

2. Ant 打包工具

3. Maven 项目构建管理

### 5.1.2 系统测试用例

如下代码段5-1是针对爬虫爬取数据的单元测试用例。改代码段完成对爬虫代理，爬虫页面处理器以及管道输出功能的集成测试。

代码5-1 页爬虫单元测试用例

|  |
| --- |
| @Test  public void test() {  try {  WeiboCrawler weiboCrawler=new WeiboCrawler();  WeiboConfig weiboConfig=new WeiboConfig();  weiboCrawler.site=weiboConfig.getSite();  weiboCrawler.weiboProcessorContext=new WeiboProcessorContext();  String url="http://m.weibo.cn/api/container/getIndex?type=uid&value=%s";  Request request=new Request(String.format(url,weiboConfig.getUid()));  Map<String, Object> pageExtrasMap=new HashMap<String, Object>();  pageExtrasMap.put("pageType",PageType.CONTAINER\_ID);  pageExtrasMap.put("uid",weiboConfig.getUid());  request.setExtras(pageExtrasMap);  Spider weiboSpider=Spider.create(weiboCrawler)  .addRequest(request)  .addPipeline(new JsonFileExPipeline(weiboConfig.getDataDir()))  .thread(weiboConfig.getThreadNum())  .setScheduler(new FileCacheExScheduler(weiboConfig.getUrlCacheDir()))  .setDownloader(new HttpClientExDownloader());  SpiderMonitor.instance().register(weiboSpider);  weiboSpider.start();  } catch (Exception e) {  e.printStackTrace();  }  } |

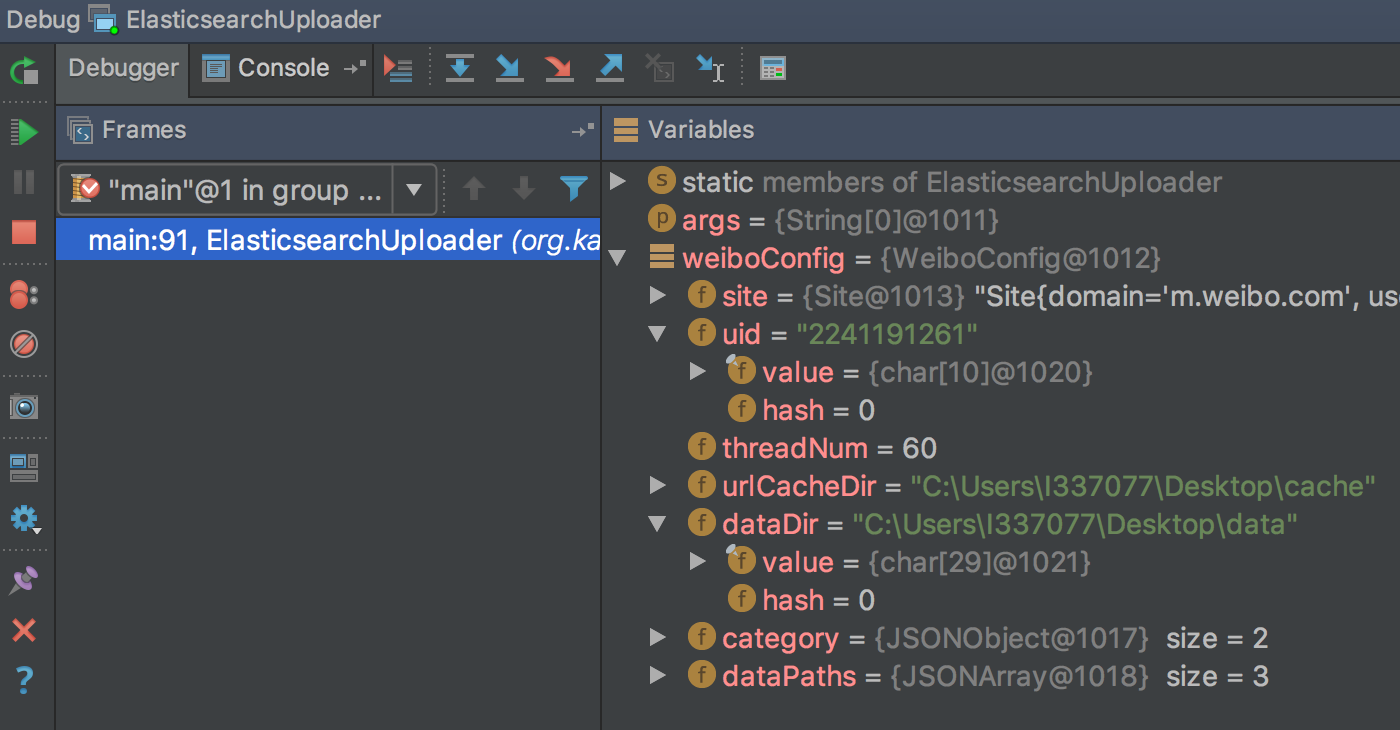
通过idea对代码进行断点调试，我们可以及时发现单元测试用例中失败用例的问题所在，如下图5-1所示时测试ElasticsearchUploader初始化时配置异常时出现问题通过调试查看变量上下文信息。

图5-1 调试获取上下文信息

## 5.2. 数据分析方法

**1．决定目标。**数据价值链的第一步必须先有数据，这些目标通常需要进行大量的数据收集和分析。因为我们正在研究数据驱动决策，我们需要一个可衡量的方式知道业务正向着目标前进，关键指标或性能指标必须及早发现。

**2．数据收集。**撒一张数据的大网，更多数据，特别是数据从不同渠道找到更好的相关性，建立更好的模型，找到更多可行的见解。大数据经济意味着个人记录往往是无用的，在每个记录可供分析才可以提供真正的价值。通过多种渠道获取需要分析的数据，不要忽略数据噪点，数据取样应该足够丰富，有说服力。

**3．数据清洗。**数据分析的第一步是提高数据质量。数据科学家处理正确的拼写错误，处理缺失数据和清除无意义的信息。在数据价值链中这是最关键的步骤，即使最好的数据值分析如果有垃圾数据这将会产生错误结果和误导。这通常意味着自动化的过程，但这并不意味着人类无法参与其中。

**4．数据建模。**数据建模是一种用于定义和分析数据的要求和其需要的相应支持的信息系统的过程。因此，数据建模的过程中，涉及到的专业数据建模工作，与企业的利益和用户的信息系统密切相关。从需求到实际的数据库，有三种不同的类型。用于信息系统的数据模型作为一个概念数据模型，本质上是一组记录数据要求的最初的规范技术。数据首先用于讨论适合企业的最初要求，然后被转变为一个逻辑数据模型，该模型可以在数据库中的数据结构概念模型中实现。一个概念数据模型的实现可能需要多个逻辑数据模型。数据建模中的最后一步是确定逻辑数据模型到物理数据模型中到对数据，访问，性能和存储的具体要求。数据建模定义的不只是数据元素，也包括它们的结构和它们之间的关系。

**5．数据可视化。**数据可视化主要旨在借助于图形化手段，清晰有效地传达与沟通信息。但是，这并不就意味着，数据可视化就一定因为要实现其功能用途而令人感到枯燥乏味，或者是为了看上去绚丽多彩而显得极端复杂。为了有效地传达思想概念，美学形式与功能需要齐头并进，通过直观地传达关键的方面与特征，从而实现对于相当稀疏而又复杂的数据集的深入洞察。然而，设计人员往往并不能很好地把握设计与功能之间的平衡，从而创造出华而不实的数据可视化形式，无法达到其主要目的，也就是传达与沟通信息。数据可视化与信息图形、信息可视化、科学可视化以及统计图形密切相关。

**6．优化和重复。**数据价值链是一个可重复的过程，通过连续改进价值链的业务和数据本身。基于模型的结果，企业将通过数据科学团队测量的结果来驱动业务。在结果的基础上，企业可以决定进一步行动通过数据科学团队提高其数据收集、数据清理和数据模型。如果企业对于重复这个过程越快，就越早能走向正确的方向，从而得到数据价值。理想情况下，多次迭代后，模型将生成准确的预测，业务将达到预定义的目标，数据价值链的结果将用于监测和报告。

## 5.3. 微博数据结果分析

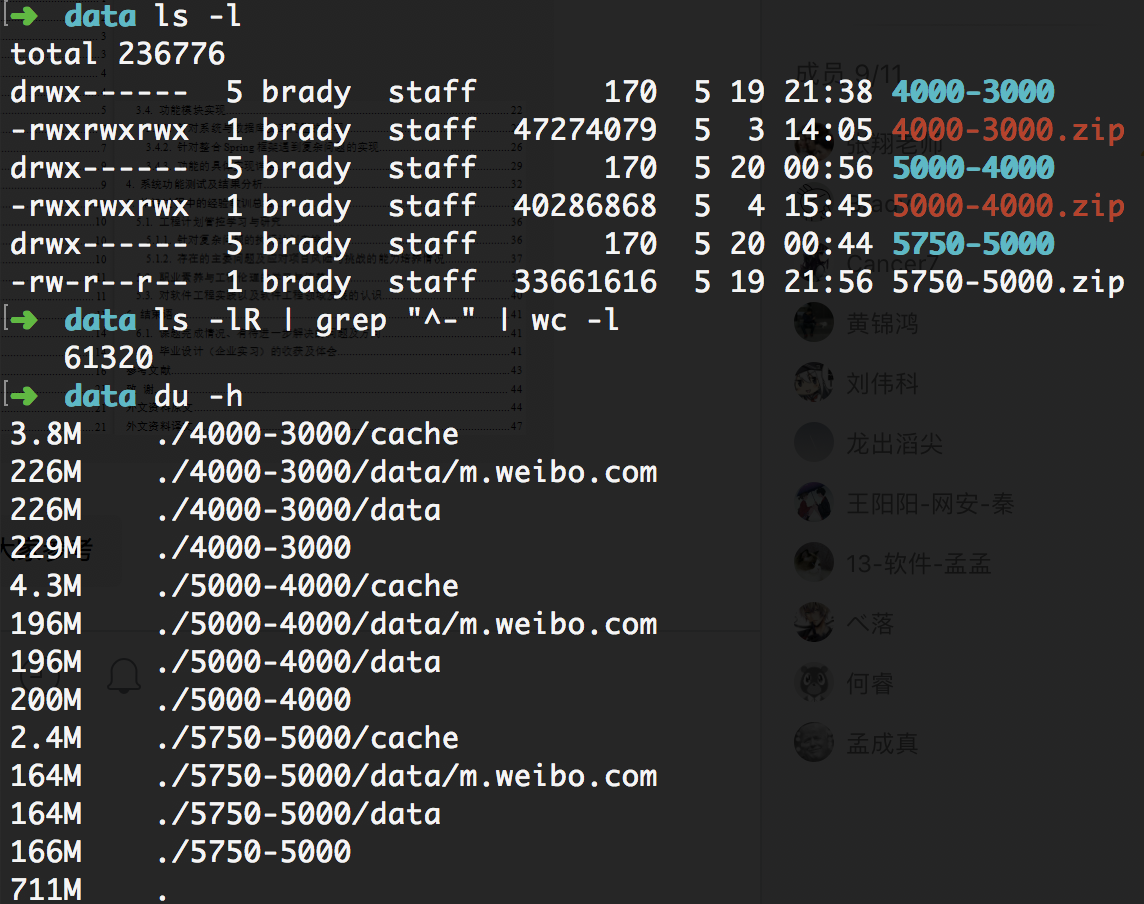
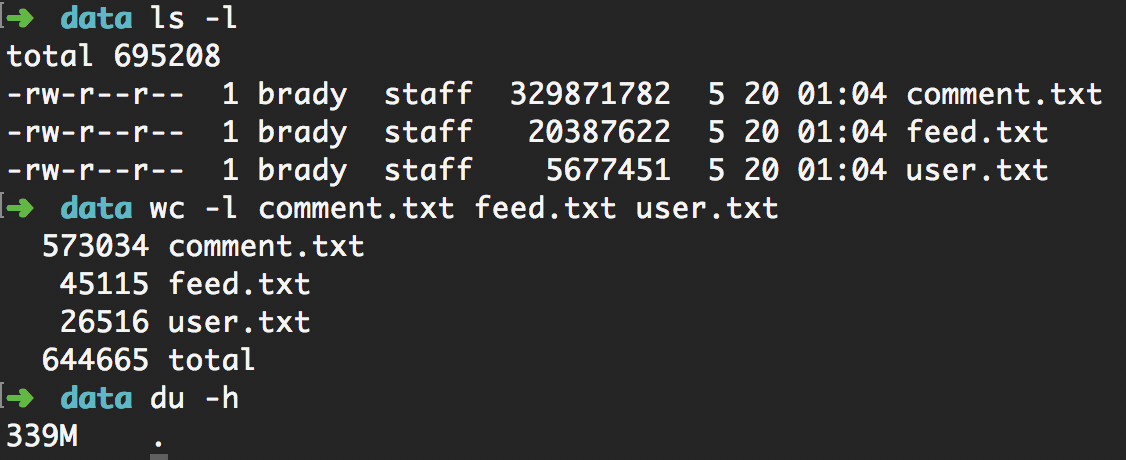
爬虫爬取页面获取的数据如下图5-2所示。通过执行ls –l可以获取数据目录下的文件情况，通过执行ls –lR|grep ‘^-’|wc –l统计出一共爬取的有效数据网页时61320个，且数据总大小为711兆。

图5-2 爬虫数据采集情况

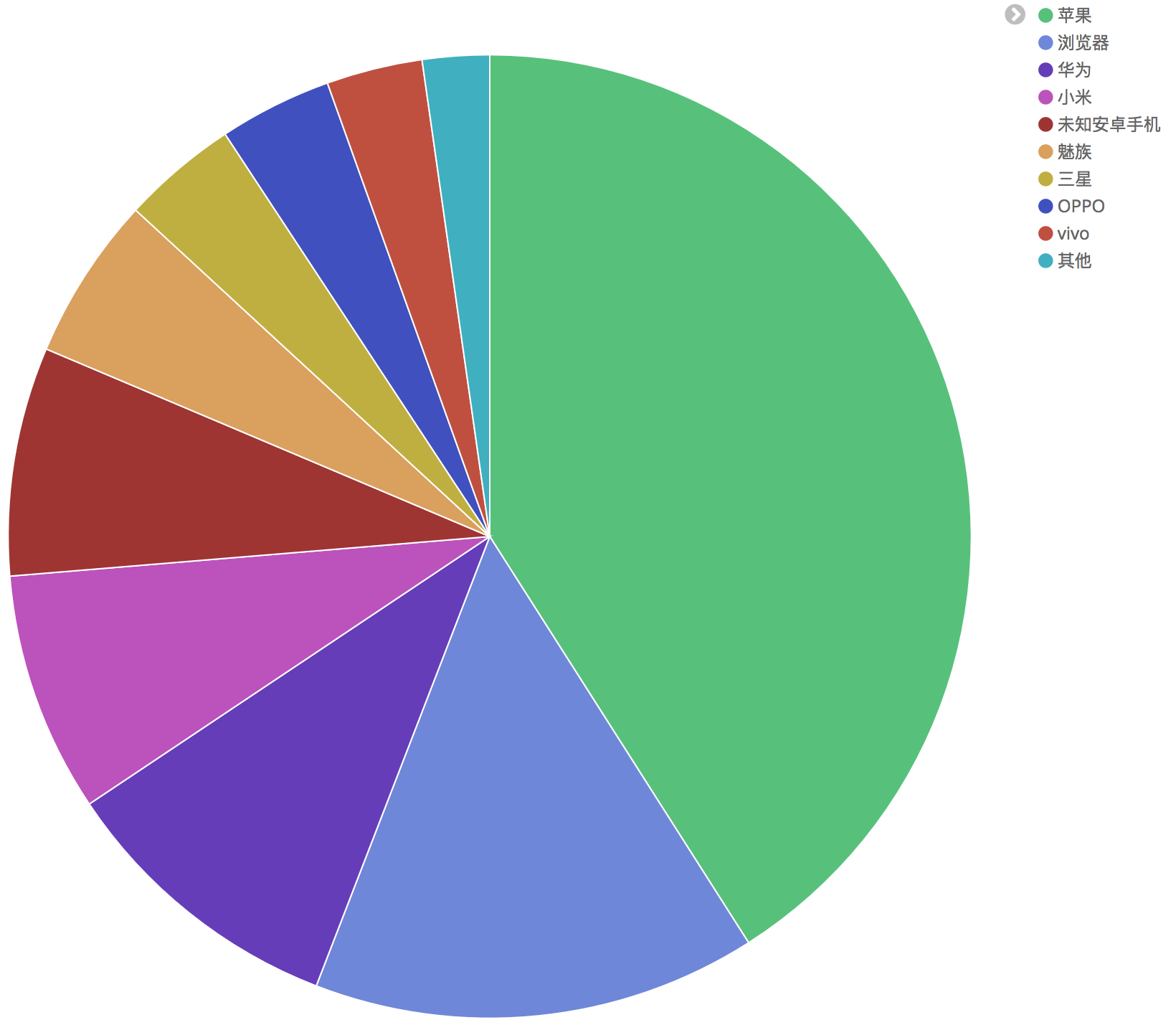
经过清洗分析后的数据情况如下图5-3所示。经过数据整理后，将数据格式化为系统设定的三种数据模型，整理后的数据大小一共339兆，有效评论数据为573034条，有效动态数据为45115条，有效用户数据为26516条，有效数据一共有644665条。

图5-3 有效数据统计

基于以上数据，我们可以通过Kibana分析出以下几种（举例，并非全部）数据特性和规律。

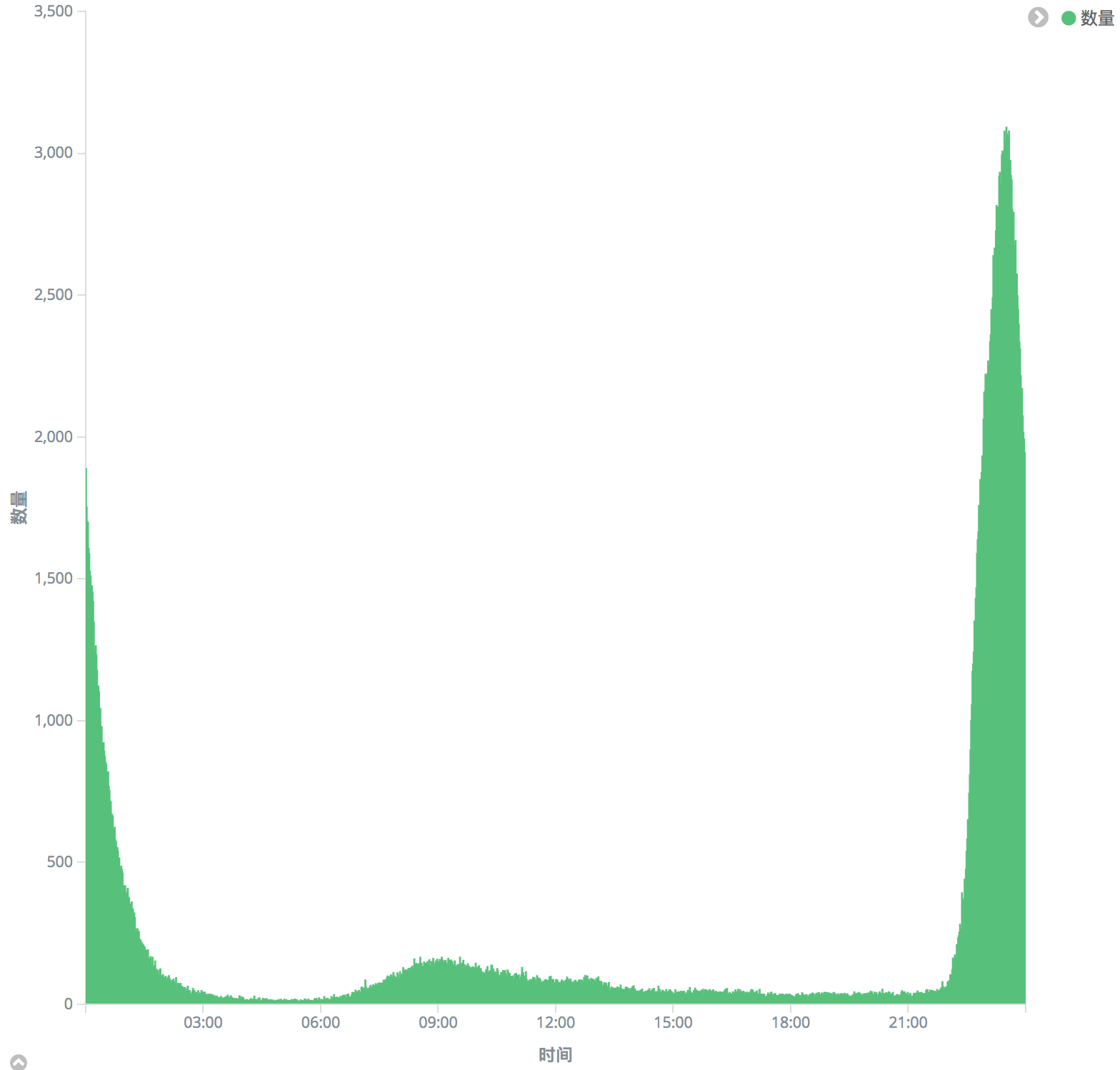
**1. 微博用户设备统计：**

由下图5-4可以看出，微博树洞中苹果用户最多，其次是来自于网页版的用户，接下来则是国产手机，由此可见，现在微博树洞用户人群特点是大部分人使用苹果手机，其实是国产手机，华为在国产手机中首屈一指，也正是说明了华为最近销量一路攀升和口碑的提高得到了学生人群消费者的认可。排名第二的为浏览器用户说明树洞中有一定数量的用户通过电脑浏览树洞。

图5-4 微博用户来源设备统计

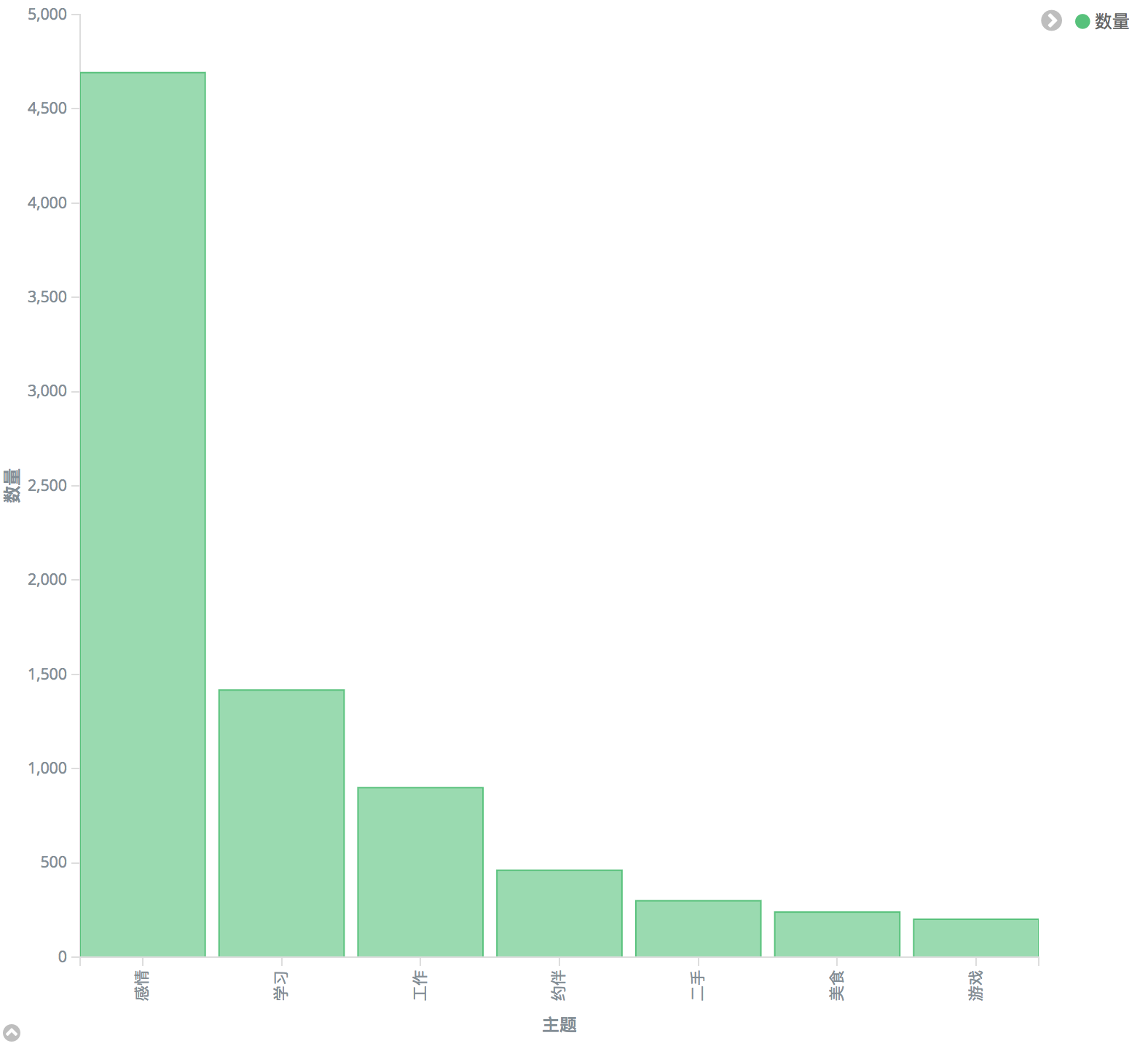
**2. 每分钟评论数量随时间变化**

由下图5-5所示，树洞评论活跃期主要集中在晚上十一点到凌晨两点，其中早上八点开始到中午一点有一段树洞评论活跃期，数据分析结果正好符合人们早上的作息时间。

图5-5 评论数量随时间变化折线图

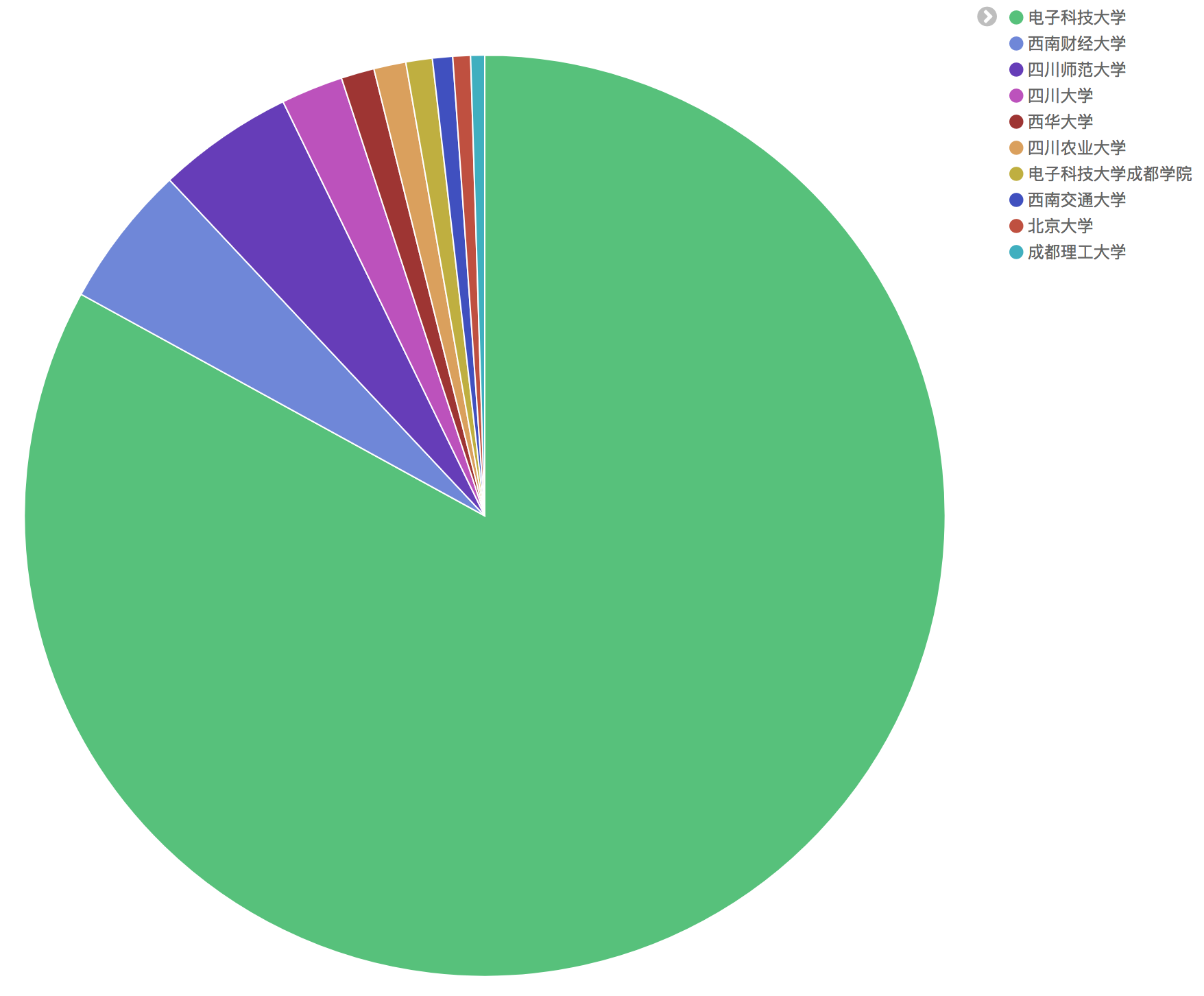
**3. 树洞主题数量统计**

经过对数据的清洗，根据对微博动态内容进行初步的语义分析，提取多个动态内容建立主题关键字数据集，最后再对全量数据进行主题分类，并且根据关键字分类统计不同主题的数量，提取了排名前十的树洞主题，绘制出微博动态各种类别数量柱状图。如图5-6所示，树洞主题中情感话题的数量最高，其次是学习和工作，游戏最末，排除感情这个和树洞本身特性相关的主题之外，说明了树洞人群大部分人还是以交流学习为主。

图5-6 微博动态主题分布

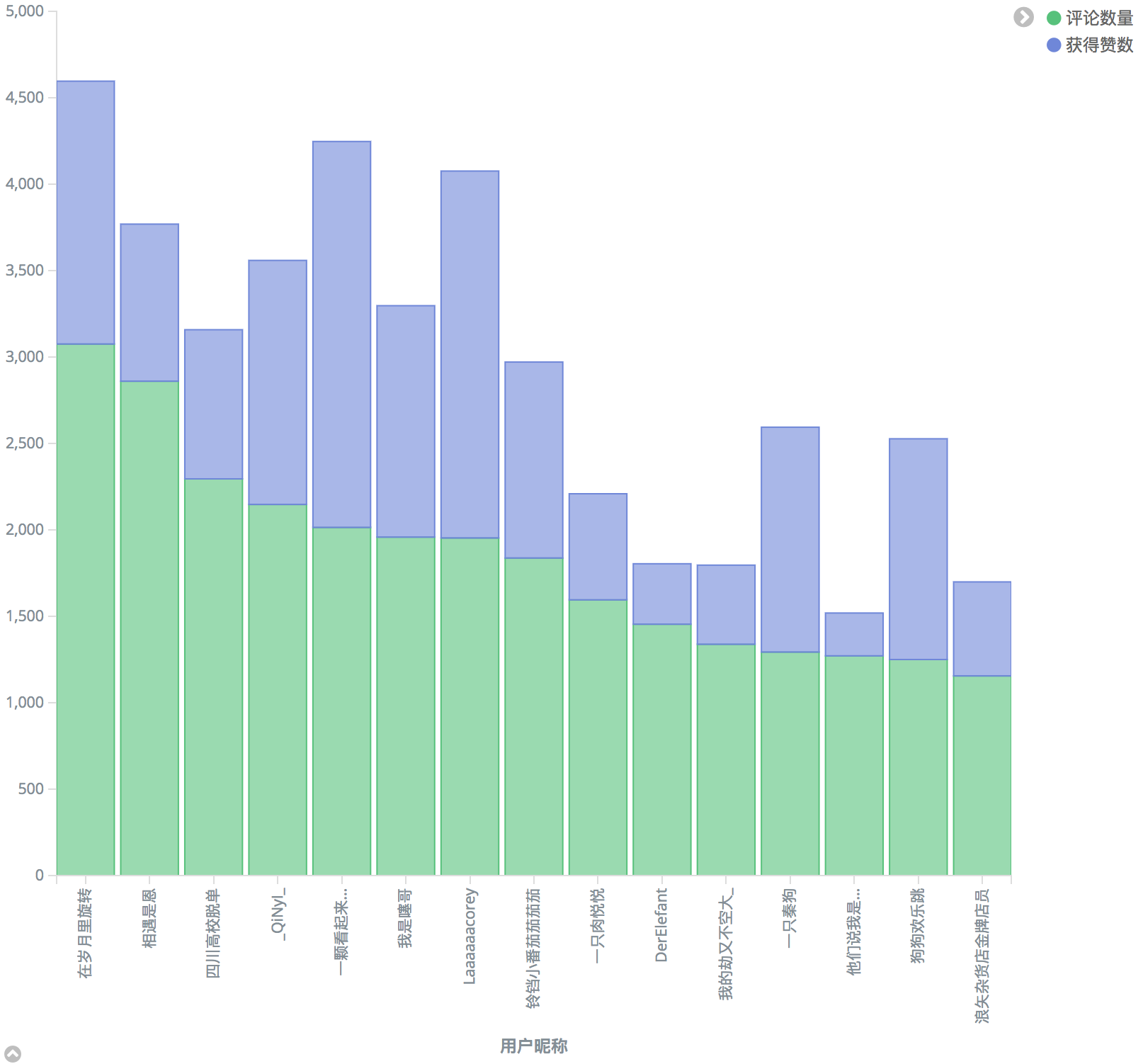
**4. 微博用户学校占比**

经过爬取微博评论的来源字段，并且对手机品牌通过关键字分析划分品牌大类，对用户的手机品牌种类进行分类统计，除去非法数据后，有效数据的统计图如下图5-7所示。可以看出树洞中用户的学校主要还是电子科技大学，其次是西南财经大学。且出了排名第九名的北京大学外，其他学校都是四川的大学，说明树洞的影响力主要在四川内的大学，北京大学应该是电子科技大学保送到北京大学的学子，因为电子科技大学每年都有一定比例的人保送到北京大学。

图5-7 微博用户学校分布

**5. 树洞用户评论情况**

如图5-8所示，展示了微博用户的评论情况和获取的赞的情况。根据平时的浏览情况，可以看出树洞里面评论数量最多的前十五名用户也是平时经常看见的用户，但是获取的赞的总数并不随评论数量增加而增加，因此可以判断赞的数量主要取决于评论的质量。

图5-8 树洞用户评论情况

## 5.4. 后续扩展研究计划

本课题只实现了基本的数据清洗分类和关键字提取分析语义，但是由于样本不够，所以语义提取不是很准确，后期可以通过增加数据量来训练更加成熟的模型，通过训练模型能够分析出更加有价值的信息。例如可以通过分析树洞动态和评论的语义情感倾向来获取树洞中的热点事件，并且可以通过分析用户之间的关联关系，对树洞人群有更精确的定位分析。通过深度学习可以挖掘出更多有价值的信息，这个也正是现目前数据科学家们的研究方向，很值得研究。

# 第六章 职业素养与工程伦理的学习与培养

## 6.1. 个人职业素养培养

在软件工程中，要解决问题，必须增强及时、敏锐发现问题的能力。问题作为一种客观存在，有的显而易见，有的深藏于复杂的表象之中，有的则通过别的形式反映出来，只有保持强烈的问题意识，才能增强洞察问题的敏锐性。问题的形成和发展，必然要具备一定的条件，经历一个时间过程，越早发现问题越小，越晚发现则问题越大，解决的难度也就越大，甚至会吃大苦头。因此，在编写代码的时候，我们就应该考虑它的封装性和可测试性。有了强烈的问题意识，才能及早发现问题。

通过在腾讯云的实习，让我解决问题的能力有了很大的提高，作为平台方，我们每天都会有大量的业务接入，电话也是接个不停，我们每天都要花费大量的时间和业务排查问题，后来业务量增加更快，这种方式显然已经不合适了，因此我们通过持续优化文档，开发自主联调定位问题工具来降低我们在业务接入中的参与度。这让我学会了从问题的本质去分析问题产生的原因，问题未来的可控性，及时发现和处理问题。现在通常遇到一个问题时，我不会像在之前一样先去急着去解决这个问题，而是现对问题做个大概的总体分析，评估其影响范围和修复可能会带来的风险，通过这种解决前评估的方式能够大大降低修改一个问题而引入其他问题的风险。

一、理论学习使我对工作有了更进一步的认识和了解。

不懂就学、就问，是一切进步取得的前提和基础，使我学习工程中对软件工程知识更为容易理解和掌握。工作中时常遇到新的问题，就需要查阅相关资料，请教同事，一个问题一个问题的解决，一个困难一个困难的克服，不仅能够将原有知识温习巩固，产生新的理解，而且学到很多新知识，有了许多新的认识。但某些认识都还是肤浅的，还需要我在实践当中去不断深入地理解。

二、实践让我的业务技能不断增长，工作能力不断加强。

刚开始工作时，对于腾讯的很多工作流都不是很熟悉，一边是同事们的悉心指导，一边是自己对着文档反复琢磨。几个月下来，已经能比较熟练地掌握基本的工作方法和一些简单的技巧，而且已经能够独立完成一些简单的接入和模块优化需求。

在完成各项工作的同时，我还存在着以下几点的不足：一是工作中有时自信心不足，反而出现低级错误;二是在处理一些事情时有时显得较为急躁;三是对一些业务技能特别是应用工程软件还不够熟悉。

在今后的工作中，我将发扬成绩，克服不足，朝着以下几个方向努力：

1、学无止镜，时代的发展瞬息万变，各种学科知识日新月异。我将坚持不懈地努力学习各种理论知识，并用于指导实践，以更好的适应行业发展的需要。

2、熟练的掌握各种业务技能才能更好的投入工作，我将通过多看、多学、多练来不断的提高自己的各项专业技能。

三、努力学习，注重实践，不断提高自身素养和工作能力

回到学校后，我将更加专注于学习计算机基础知识，在工作遇到的很多性能优化问题，都需要扎实的计算机基础知识来做支撑，比如数据库查询优化问题，如果对于数据库索引的实现原理不熟悉的话很难正确高效的使用索引提高查询性能。

二、围绕中心，突出重点，尽心尽力履行职责

进入公司后，在各项工作中我都本着认真负责的态度去对待每项工作，受到了同事们的好评。一是为使自己尽快熟悉工作，进入角色，我一方面抓紧时间查看相关文档，熟悉自己的工作职责，另一方面我虚心向公司的领导、同事请教使自己对公司的情况有了一个比较全面、系统的认识和了解。不将自己的私人感情和与同事的关系带到工作中，经过一段时间的努力我的工作得到了大部分同事的理解和支持，和同事们相处融洽。

## 6.2. 软件工程对社会的影响

软件工程的发展有两大趋势，首先是全球化。Internet技术无疑是20世纪末伟大的技术之一,它为我们提供了一种全球范围的信息基础设施，随着经济全球化的发展，导致了信息全球化，这必然会促使软件工程朝着全球化协作开发 ，很多知名的企业很早之前就开始在全球寻找优秀人才，扩充着自己的实力，充分利用全球资源。比如说印度的软件开发就是软件外包行业的一个充分体现。还有很多外企，比如说IBM，经常会看到一个软件在美国进行概念设计，在欧洲实现系统架构的搭建，在中国实现软件的编码测试，最后由印度负责软件的售后支持。

第二点是开放性，这里面包含了开放标准、开放架构、开源软件三个方面。统一的标准， 使得不同企业开发的不同软件可以实现相同的接口，即可以相互调用，举个简单的例子，电脑里面的电子配件千千万万，都是由不同厂商之间制造的，同一个电脑主板，可以安装不同厂商的电脑内存条、显卡、声卡等，是因为他们实现了相同的接口，即统一的标准。开放的架构和开源的软件有些类似，都是别人劳动的智慧的结晶。不仅可以直观看到别人的代码和架构，还可以了解到别人解决问题的思想，从中受益得到启发。通过这些开放性能够实现互通互联，不同的文化和国界将可以更好地交流协作，软件用户的投资将得到非常好的保障。

虽然软件工程的发展为我国甚至人类做出了无与伦比的贡献，但近年来软件工程的迅猛展也带来了许多负面效应。其中最突出的就是人与人和人与社会之间的矛盾冲突日益严峻。2011年互联网用户信息泄密事件，使软件工程的两面性问题又一次成为人们关注的焦点。

人们在享受软件工程技术发达带来的好处同时，也感到了它对自身的威胁，凡是有使用的地方，就会有安全问题存在。

首先是人与人矛盾，表现在网络诈骗以及网络黑客等诸多方面，随着科学技术的迅猛发展，越来越多的人们开始使用网络通信，很多诈骗者通过各种方法盗窃QQ账号、邮箱账号后，向用户的好友、联系人发布信息，声称遇到紧急情况，请对方汇款到其指定账户。造成人们精神以及财产的损失。

第二点是人与社会的矛盾加剧，软件工程的发展对人与社会的关系造成的负面影响，比如“网络谣传”的泛滥，让人与人之间的信任降到了零点。7.23温州动车事故，造成了生命不可挽回的惨剧。还有很多，比如恶意软件和病毒软件，人们在享受科学技术带来快捷便利的同时，感受到了计算机网络安全和用户隐私构的威胁。

最后一点是人与自然间的矛盾，科学技术的快速的发展，使得软件工程技术和硬件技术相互促进共同进步，也就造成了很多电子设备的淘汰，这些电子设备往往包含了很多的重金属，虽然现在有很多的回收机制，但还是无法避免的造成了浪费。电子设备更新换代的越来越快，更快更新更好的产品一旦出现在市场，往往宣告了以往电子设备的生命周期的结束。比如说我们身边的手机，几年前我们用的按键手机，现如今我们用的触摸屏幕手机，手机屏幕越来越大，可用软件越来越多，手机对性能的需求也在不断提升，所以我们一直不断的换手机，而换下的手机往往被回收的很少。直接导致不可再生的资源趋于枯竭，新设备的生产还造成了更多的环境的污染，包括大气污染、水污染以及更多的固体废弃物的污染。软件工程的发展，给人类社会带来了空前的繁荣，也带来的很多负面影响。

# 第七章 对软件工程实践以及软件工程领域发展的认识

## 7.1. 软件工程和可持续发展

做好代码规范、提高代码质量，能显著增强代码的可读性、可维护性和可变更性。本文将这三大要素合称为代码的读写可维护性，努力提高代码的读写可维护性，是做好代码规范的必要非充分条件。代码规范和架构设计是软件的灵魂所在，代码质量偏低，就像是人失去了三魂七魄中的一魄，就会丧失活力，影响正常运行，增加软件交付后维护成本，出现推迟完成、超出预算、特性缺失等现象。

因此公司有必要制定或修订自己的代码规范，以确保规范本身不存在太大疏漏，而后加强执行力度，确保程序员自觉编写符合代码规范的编码产品。

每一种语言都有各自的特性，因此需要对每一种语言单独制定代码规范。但同一家公司的多份代码规范不能差异过大，应在兼顾语言风格和官方编码指南的基础上进行统一。

代码的可读性是指代码让人容易阅读、跟踪和理解的程度。提高代码的可读性可以为代码阅读者节约时间（避免阅读时浪费过多无谓的时间）和精力（Debug、扩展功能或是性能优化的前提条件是你要读懂这段代码）。编码风格一致，代码清晰表达意图。不在代码中添加不必要的功能，避免做没有价值的工作，坚持操作方法的原子性，而后使用组合模式实现业务逻辑，避免大段代码，要写高内聚、低耦合的代码。

因此软件工程的可持续发展很大程度上依赖于代码的可维护性，只有严格遵守合适的编码规范，我们的系统才是易于维护的。

## 7.2. 软件工程发展状况

(1) 第一阶段：瀑布模型。软件的工业化和类生命性最终导致了瀑布模型的出现，瀑布模型构建了软件世界中一个最基本的生命周期理念。

(2) 第二阶段：演化过程模型。如同生物的演化并非一蹴而就，成熟的软件也应是通过多个细小的生命循环周期迭代而成，演化过程模型的出现再次验证了软件所特有的类生命特征。

(3) 第三阶段：统一过程阶段。 统一过程模型和CMM/CMMI的提出，反映了人类的一个基本需求，即在高速变化的环境中试图追求内部的相对稳定，这同时也标志着软件工程进入到相对成熟和稳定的阶段。

(4) 第四阶段：惯例模型和敏捷过程模型并存阶段。敏捷过程反映了人类的另一个基本需求—随变顺变应变。当软件工程演化到这一阶段时，如同生物界和计算机世界演化一样，实际上已经开始显现种群化的特征。

(5) 第五阶段：模型驱动和云计算阶段。模型驱动是为了满足软件工业自动化的需求，云计算则基于Internet，强调面向服务与计算资源整合。这两个概念提出已久，而真正启动才刚刚开始，距离成熟有待时日。但也进一步证明了软件工程开始进入种群化阶段。

需要指出的一点是，由于软件工程的发展遵循链式原则，软件工程的发展也总是滞后于软件的发展，通常是在软件的某项语言、技术或工具进入稳定期后，软件工程才会有相应新的思想出现。

例如，软件的高级语言在1954年已出现，而软件工程在1968年才刚刚提出；当软件进入结构化程序阶段时，瀑布模型刚刚显现；当软件面向对象技术逐渐成熟时，演化过程模型开始出现；当面向对象技术开始和网络紧密结合时，统一和敏捷过程才开始出现。

可以预见的是，未来软件发展和软件工程发展之间的间隔会逐步缩短，5－10年的滞后期有可缩短到2－3年。

# 第八章 结束语

六个月的实习很快就过了，在SAP的六个月时间学到的东西也是十分的丰富，不仅仅是技术上有所提高，在待人处世和与人沟通上面也有很大的提高。

记得第一天来这里时，心里不可避免的有些疑惑：不知道老导师怎么样，应该去怎么做，要去感谢什么等等。坐在工位后，之间几个陌生的人用莫名而疑惑的眼神看着我，我微笑和他们打招呼，尴尬的局面理科得到了缓解，大家都很友善的微笑欢迎我的到来。这让我觉得这是 一个很温馨的团队，后来的六个月相处还发现除了温馨，他们还很有趣。

另外一点就是感受到学校和社会的距离，在学校，只有学习的氛围，毕竟学校是学习的场所，每一个学生都在为取得更高的成绩而努力。在这里是工作的场所，每个人都会为了获得更多的报酬而努力，无论是学习还是工作，都存在着竞争，在竞争中就要不断学习别人先进的地方，也要不断学习别人怎么做人，提高自己的能力。要与社会交流。为社会做贡献。只懂得纸上谈兵是远远不及的，以后的人生旅途是漫长的，为了锻炼自己成为一名合格，对社会有用的人才，多接触社会是很有必要的。

在短暂的实习过程中，我深深的感觉到自己所学的知识的肤浅和在实践运用中知识的匮乏，刚开始的一段时间里，对一些工作无从下手，茫然不知所措，这让我感到非常的难过。在学校总以为自己学的不错，一旦接触到时间，菜发现自己知道的是多么少，这才真正领悟到学无止境的含义。在以后的工作中，我将用更多的时间去学习基础知识和基本理论，为以后的发展打好基础。

# 参考文献

[1] Gupta, Yuvraj，Kibana Essentials，Packt Publishing.，2015-10-30

[2] 饶琛琳，ELK stack 权威指南，机械工业出版社，2015-10-01

[3] Richard Lawson，用Python写网络爬虫，人民邮电出版社，2016-08-01

[4] 王晶 朱珂 汪斌强，基于信息数据分析的微博研究综述，ISSN 1001-9081

[5]王鲁飞，基于新浪微博数据的处理与用户行为分析，北京交通大学，2016-06

[6] Wes McKinney，Python for Data Analysis，机械工业出版社，2014-01-01

[7] Michael Milton，深入浅出数据分析，电子工业出版社，2013-04-01

[8] Stolcke，SRILM - An Extensible Language Modeling Toolkit，2002

[9] Kaufmann and J. Kalita， Syntactic Normalization of Twitter Messages，2010

# 致 谢

转眼间，在电子科技大学四年的本科学习生活就要结束了，而入学仿佛还是昨天的事情，初来乍到时的场景犹历历在目。回忆起这四年的点点滴滴，感慨不已，欣慰之余而又庆幸无比。值得欣慰的是，我这四年的时间是在汗水和拼搏中度过的，通过和同学老师一起实施项目，不仅加强了我对理论知识的理解，也增强了我的实践运用能力，这使我学到了许多受益无穷的东西。对我以后都有着深远的影响。

本论文是在张翔老师和企业导师范家兵导师的悉心指导下完成的。从选题、到构思、再到撰写，一直到最终的定稿，张老师和我的企业导师给了我很多的帮助，张老师和我的企业导师对学术的钻研精神和严谨的治学态度，都值得我终生学习，在张老师和我的企业导师的悉心指导下，我的毕业论文顺利完成，在此对他们表示由衷的感谢。此外，我还要感谢在一起愉快的度过大学生活的每个可爱的同学们和尊敬的老师们，正是由于你们的帮助和支持，我才能克服一个一个的困难和疑惑，直至本文的顺利完成。还要感谢在毕业设计过程中为我提供帮助的同学。本人在撰写论文的过程中，参阅了大量的文献资料，在此，谨向相关作者、编者、出版社表示感谢。

在电子科技大学学习和生活的四年里，每一位任课老师都尽心尽责，在此，衷心的感谢各位老师四年来的悉心栽培。同时，四年的时光，我结识了许多同学，分别在即，我想借此机会感谢你们四年来的陪伴，愿同窗之谊永续长存。最后，感谢我的母校电子科技大学为我提供了良好的学习环境和迈向人生下一阶段的优秀平台。在今后的学习和工作中，我必将时刻铭记电子科技大学“求真求实，大气大为”的校训，时刻不忘作为一名电子科技大学毕业生应该有的责任和担当，为祖国的计算机科学的发展和实践奉献自己的一份力量。

# 外文资料原文

## Sentiment Analysis of Twitter Data

Apoorv Agarwal Boyi Xie Ilia Vovsha Owen Rambow Rebecca Passonneau

Department of Computer Science

Columbia University

New York, NY 10027 USA

Sentiment analysis has been handled as a Natural Language Processing task at many levels of granularity. Starting from being a document level classi- fication task (Turney, 2002; Pang and Lee, 2004), it has been handled at the sentence level (Hu and Liu, 2004; Kim and Hovy, 2004) and more recently at the phrase level (Wilson et al., 2005; Agarwal et al., 2009).

Microblog data like Twitter, on which users post real time reactions to and opinions about “everything”, poses newer and different challenges. Some of the early and recent results on sentiment analysis of Twitter data are by Go et al. (2009), (Bermingham and Smeaton, 2010) and Pak and Paroubek (2010). Go et al. (2009) use distant learning to acquire sentiment data. They use tweets ending in positive emoticons like “:)” “:-)” as positive and negative emoticons like “:(” “:-(” as negative. They build models using Naive Bayes, MaxEnt and Support Vector Machines (SVM), and they report SVM outperforms other classifiers. In terms of feature space, they try a Unigram, Bigram model in conjunction with parts-of-speech (POS) features. They note that the unigram model outperforms all other models. Specifically, bigrams and POS features do not help. Pak and Paroubek (2010) collect data following a similar distant learning paradigm. They perform a different classification task though: subjective versus objective. For subjective data they collect the tweets ending with emoticons in the same manner as Go et al. (2009). For objective data they crawl twitter accounts of popular newspapers like “New York Times”, “Washington Posts” etc. They report that POS and bigrams both help (contrary to results presented by Go et al. (2009)). Both these approaches, however, are primarily based on ngram models. Moreover, the data they use for training and testing is collected by search queries and is therefore biased. In contrast, we present features that achieve a significant gain over a unigram baseline. In addition we explore a different method of data representation and report significant improvement over the unigram models. Another contribution of this paper is that we report results on manually annotated data that does not suffer from any known biases. Our data is a random sample of streaming tweets unlike data collected by using specific queries. The size of our hand-labeled data allows us to perform crossvalidation experiments and check for the variance in performance of the classifier across folds.

Another significant effort for sentiment classification on Twitter data is by Barbosa and Feng (2010). They use polarity predictions from three websites as noisy labels to train a model and use 1000 manually labeled tweets for tuning and another 1000 manually labeled tweets for testing. They however do not mention how they collect their test data. They propose the use of syntax features of tweets like retweet, hashtags, link, punctuation and exclamation marks in conjunction with features like prior polarity of words and POS of words. We extend their approach by using real valued prior polarity, and by combining prior polarity with POS. Our results show that the features that enhance the performance of our classifiers the most are features that combine prior polarity of words with their parts of speech. The tweet syntax features help but only marginally.

Gamon (2004) perform sentiment analysis on feedback data from Global Support Services survey. One aim of their paper is to analyze the role of linguistic features like POS tags. They perform extensive feature analysis and feature selection and demonstrate that abstract linguistic analysis features contributes to the classifier accuracy.

# 外文资料译文

## Twitter数据的情感分析-节选

Apoorv Agarwal Boyi Xie Ilia Vovsha Owen Rambow Rebecca Passonneau

计算机科学系

哥伦比亚大学

纽约，NY 10027 USA

情感分析已被处理作为一个自然语言处理任务在许多层次的粒度。从一个文档级分类分类任务（Turney，2002；庞和李，2004），它已在句子层面的处理（胡和刘，2004；基姆和Hovy，2004）和最近在短语层面（Wilson et al.，2005；Agarwal et al.，2009）。

像Twitter这样的微博数据，用户在实时反应和对“一切”的意见上都提出了新的和不同的挑战。Twitter数据的情感分析的早期和最近的一些结果是由Go et al.（2009）、（2010 Bermingham和Smeaton）和Pak（2010）和帕劳贝克。等人（2009）利用远程学习获取情感数据。他们用积极的表情像“微博结尾：）”“：-）”的积极和消极的表情符号“：（”“：-（”为负。他们建立模型，使用朴素贝叶斯，马克桑和支持向量机（SVM），他们报告的SVM的性能优于其他分类。在特征空间上，他们以一个单字，与词类结合bigram模型（POS）的特点。他们注意到一元语法模型优于其他模型。具体而言，二元语法和POS功能没有帮助。巴基斯坦和帕劳贝克（2010）通过远程学习模式收集数据。他们通过主观和客观两种方式执行不同的分类任务。主观数据收集以表情结束的动态内容。客观数据抓取公众传媒的推特账户如“纽约时报”、“华盛顿帖”等，他们报告指出，POS和二元语法对于情感分析均是有一定的作用的，这个结果和Go et al提出的结果相反，这些方法主要是基于N-gram模型。此外，他们使用的数据进行训练和测试收集的搜索查询。相比之下，我们目前的功能，实现对一个单字的基线显著增加。此外，我们探索了一种表示数据和报告重大改进统计模型的不同方法。本文的另一个贡献是，我们报告的结果手动注释的数据，不遭受任何已知的偏差。我们的数据是一个随机抽样的不同的数据收集使用特定的查询。我们手上标记的数据允许我们进行交叉验证实验和检查分类器在折叠中的性能变化。

Twitter数据情感分类的另一项重大努力是由巴博萨和冯（2010）。他们使用极性预测从三个网站作为标签训练模型，并使用1000个手动标记的数据来调整模型和另外1000个手动标记的数据来测试。然而，他们并没有提到他们如何收集他们的测试数据。他们提出推特动态，标签，链接的语法特征，在特征词和词性词语之前的连接的标点符号和感叹号。我们的结果表明特征提高了我们分类器的性能，推特的语法分析功能在也为性能的提高提供了一点帮助。

Gamon（2004）反馈数据进行情感分析的全球支持服务调查。他们的论文的目的之一是情感分析功能，如POS标签的语言功能。他们进行了广泛的功能分析和特征选择和证明了抽象的语言分析功能有助于分类精度。