武汉理工大学毕业设计（论文）

**人脸跟踪与识别系统实现**

学院（系）： 计算机科学与技术学院

专业班级：软件工程专业 软件zy1202班

学生姓名： 杨晨

指导教师： 岑丽

学位论文原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包括任何其他个人或集体已经发表或撰写的成果作品。本人完全意识到本声明的法律后果由本人承担。

作者签名：

年 月 日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保障、使用学位论文的规定，同意学校保留并向有关学位论文管理部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权省级优秀学士论文评选机构将本学位论文的全部或部分内容编入有关数据进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于1、保密囗，在 年解密后适用本授权书

2、不保密囗 。

（请在以上相应方框内打“√”）

作者签名： 年 月 日

导师签名： 年 月 日

摘要

在这个互联网飞速发展的时代，计算机技术已经深入到每个人的生活中。而计算机图形图像技术在生活中被也越来越多的运用起来。人脸跟踪与识别系统，涉及到人脸的检测与人脸的识别。将其运用到Android系统之中，从而实现了通过手机摄像头识别人脸的功能。

本文首先阐述了人脸检测Haar分类器、人脸识别LBP算法的相关内容，然后根据实际的Android开发内容编写了需求报告。首先通过JavaCV中Camera相关类获取视频流的一帧，然后使用Haar分类器进行人脸检测，获取人脸矩阵信息。对录入的人脸进行训练后再使用LBP算法进行人脸识别，最后在屏幕上绘制出人脸的矩阵与识别出的人的姓名。在下一帧中再进行如上的操作，从而实现了对视频流的人脸识别。

关键词：人脸识别；haar分类器；LBP算法；Android

**Abstract**

In this era of rapid development of Internet,computer technology has been deep into everyone’s life.The computer graphics technology is more and more used in our life.Face tracking and recognition system,related to face recognition and face detection.As it applied to Android system,enabling moblile phone camera face recognition function.

The paper describes the face detection Haar classifier,the content LBP face recognition algorithm,and then write a report based on the actual needs of the Android development content.Get through the first JavaCV Camera Class a video stream,and then use the Haar classifier for face detection,face matrix to obtain information.On the entry face training before using LBP face recognition algorithm,and finally draw the human face of the matrix and identified the person’s name on the screen.In the next frame,then the above operations,in order to achieve the recognition of the video stream.

**keywords**：face recognition；haar classifier；LBP algorithm；Android

目录

第1章 绪论1

1.1 研究背景及国内外研究现状1

1.2 研究目的及意义3

1.3 研究内容及目标4

1.4 论文组织结构4

第2章 相关技术及开发工具简介6

2.1 搜索引擎6

2.2 RSS介绍6

2.3 开发工具7

2.4 开发技术7

第3章 新闻搜索引擎需求分析9

3.1 功能需求9

3.1.1 爬取子系统功能需求9

3.1.2 索引子系统功能需求10

3.1.3 web客户端功能需求10

3.2 非功能需求11

第4章 系统数据库与详细设计12

4.1 数据库设计12

4.1.1 实体属性图12

4.1.2 数据库设计12

4.2 系统详细设计13

4.2.1 爬虫子系统模块设计13

4.2.2 索引子系统模块设计14

4.2.3 web客户端模块设计15

第5章 系统实现17

5.1 系统总体结构17

5.2 爬虫子系统实现流程17

5.3 索引子系统实现流程19

5.3.1 关键词提取算法分析19

5.3.2 搜索引擎排序算法分析21

5.3.3 建立索引的实现流程23

5.4 搜索引擎框架搭建25

5.4.1 B/S架构设计25

5.4.2 UI设计25

5.4.3 前端框架开发流程25

第6章 关键问题及解决方法27

6.1 爬虫子系统部分27

6.2 索引子系统部分27

6.3 web客户端部分27

第7章 总结与展望29

参考文献30

致谢32

# 

# 第1章 绪论

## 1.1 研究背景及国内外研究现状

在最近10年中，计算机科学技术取得了巨大的发展，人们的生活水平得到了很大的提升，人们越来越多的关注社会信息。在很多行业中，对人员进行的信息进行确认与身份的辨别的需求越来越大。例如公安机关设置摄像头，来监控地铁，商场，火车站等人流量打的地方，根据画面中出现的人脸来寻找犯罪嫌疑人。在以前这样的工作通常是由人工完成的，不仅消耗很多的人力，也消耗很多的时间；在机场，登机时，工作人员会对将人脸与身份证进行核对；在银行取钱时，银行需要对取钱的用户进行人脸识别，判断其是否是本人；在海关，同样也需要人脸识别技术来确认出入境人员是否为本人，这些在生活中切实存在的需求极大的推动了近些年人脸识别技术的飞速发展与应用。

人脸识别技术是通过计算机图形学，提取人脸的特征，通过这些特征，与训练好的样本进行对比，从而进行身份验证的一种技术。人脸特征与人身上其他的特征一样，例如指纹和虹膜，都具有唯一性和不容易被伪造的特性，这是作为身份鉴别依据的前提；对比其他生物特征的识别技术，人脸识别技术在操作上更简单，结果更直观，更加隐蔽。因此，在刑事案件的侦破，信息安全领域等都有广泛的应用前景。

人脸识别技术在最早的时候限制很大，只能对单一背景的正面灰度图像进行识别，也就是只能在二维中进行人脸识别。随着对多种姿态下人脸识别的研究，包括正面和侧面，人脸识别正在向三维方向发展，在维度提升的同时，识别率也在逐步提高。在这个过程中，人脸识别技术越来越成熟，但是还是存在着很多的问题，比如在复杂的背景下，系统很难识别出其中的人脸，跟踪也存在比较大的问题。人脸识别技术并非纯粹的数字技术，它需要研究人员拥有计算机图形学、计算机视觉、生物特征技术等学科的知识，是一个融合了多种学科的技术，因此对研究人员有非常高的要求。环境和人脸本身都有很大的不确定性，在环境方面，光照和图像采集工具会对采集到的图像有比较大的影响，而人脸本身由于表情，身体姿态和脸上戴的物件的不同，会对人脸识别造成影响。综上所述，人脸识别是非常有挑战性的课题。

人脸识别有悠久的研究历史。高尔顿分别在1888年和1910年在《Nature》杂志上发表了两篇关于利用人脸来识别身份的文章。这两篇文章对人类自身如何进行人脸的识别进行了分析。在当时的科学文化环境下，并没有涉及到人脸的自动识别。

1965年陈和布莱德索在Panoramic Research Inc.发表了关于自动人脸识别的研究论文。布莱德索建立了一个半自动的人脸识别系统。该系统识别人脸是以人脸的特征点的间距和比率等参数特征。这种思想也成为之后一段时间的主流。这种人脸识别思想的特点有：①将人脸拆分成若干个部件，作为特征进行识别，主要利用的是各个部件的信息和部件之间的几何关系。这是一种很直观的方法，对人脸图像有很高的要求。如果人脸图像中的人脸不是正面照或者表情有变化的话识别率就会很低。为了弥补这种方法的缺点，后来出现了更好的人脸识别方法。根据样本库中的样本和需要识别的人脸的灰度图的比较来对人脸进行鉴别，这种方法叫做模板匹配方法。②人脸识别对环境的要求非常高，在单一背景或者没有背景的情况下，人脸的位置非常容易获得，相反，在复杂的背景下，人脸的位置非常难获得。在这段时间，人脸识别由于这个原因，并没有被应用到现实场景当中。

1964年至1990年是人脸识别的第一个阶段，这一阶段人脸识别通常作为一个普通的模式识别问题来研究。这个阶段采用的技术方案是基于人脸几何特征的方法。研究者们花费了很多时间在对面部剪影曲线的结构特征的提取和分析方面。而人工神经网络也曾被用于人脸识别的问题中。早期，除了布莱德索在进行自动人脸识别之外还有戈登斯泰因、金出武雄等。金出武雄是人脸识别领域较活跃的人物，它在1973年，于京都大学完成了自动人脸识别方面的博士论文。如今，他成为了卡内基-梅隆大学机器人研究院的教授。

第二个阶段是1991年至1997年。在短短的8年时间内，人脸识别的研究进入了高潮期，取得了很多的成就。许多具有代表性的人脸识别算法在段时间内诞生。比如著名的FERET人脸识别算法，麻省理工学院媒体实验室的特克和潘特提出的“特征脸”方法。“特征脸”方法是这一时间段内的代表作，衍生出了之后许多的算法。如今特征脸已经与归一化的协相关量方法一起成为人脸识别的性能测试基准算法。麻省理工学院人工智能实验室的布鲁内里和波基奥进行了基于结构特征的方法与模板匹配方法的对比，得出了模板匹配要优于基于特征的方法这个结论，促进了基于表现的线性子空间建模和基于统计模式识别技术的人脸识别方法的发展。这一时期主要的成果有：贝尔湖米尔等提出的Fisherface人脸识别方法；麻省理工学院的马哈单提出的基于双子空间进行贝叶斯概率估计的人脸识别方法。

第三个阶段是从1998年至现在。为了解决在光照，姿态等非理想的情况下人脸识别鲁棒性较差的问题，研究者么注重研究了这一部分。吉奥盖迪斯等人提出基于光照锥模型的多姿态、多光照条件人脸识别方法。布兰兹和维特等提出了基于3D变形模型的多光照和姿态条件人脸图像分析与识别的方法。它是基于合成的分析技术。这种方法使得人脸可以用简单的举证特征作为特征，将大量弱分类器组合成了强分类器，并且使用联级技术提高检测速度，为现在实时人脸识别打下了很好的基础。沙苏哈在2001年提出了一种基于熵图像的人脸图像识别和绘制技术。为解决光照问题提供了一种重要的思路。

现在人脸识别的算法主要分成3种。

(1)基于肤色划分的人脸检测方法。首先确定肤色模型，检测出其中的肤色区域，获得可能存在的人脸区域。之后是对人脸区域的检测，通过区域特征的方法对人脸区域进行人脸检测，从而区分出具有类肤色的其他物体。这种方法在复杂背景的环境下会有比较大的误差。在复杂背景下，人脸区域可能和其他肤色区域混杂在一起，从而造成混淆，在获得肤色区域后任然无法判断出此区域是人脸区域。另一种情况是光照和面部表情造成的影响。光照和面部表情会对人脸检测造成影响，区域特征会受到影响，在这种情况下会使用聚类，归并，验证的方案来减少其带来的影响。首先将人脸肤色像素按照严格的几何关系去拆分，再将其按照一定规则合并到一起，在合并过程中使用一些其他特征进行验证。

(2)基于人工神经网的方法。由于人脸轮廓的复杂性，人脸很难用数学模型表示，而神经网是一种可以表示复杂模型的方法。人工神经网的优势是方便建模和较高的鲁棒性。但是人工神经网在运算速度上有比较大的欠缺，这是制约它发展的一个因素。

(3)基于启发式模型的方法。这种方法主要是通过抽取图像的若干特征进行人脸检测。首先对人脸局部的特征进行判断，再对人脸整体布局进行判断。也就是从人的五官的检测到人的五官的相对位置的检测。使用人脸五官分布特征的知识模型进行检测。由于抽取的特征较少，所以这种方法有比较快的检测速度。这种方法的主要障碍也是由于抽取特征较少，在复杂环境下的人脸检测成功率较低。

## 1.2 研究目的及意义

人脸识别在生活中有广泛的应用场景。比较常见的有安检，监控，身份认证等。随着人脸识别技术渐渐成熟，人脸识别会更多的出现在我们生活中的各个角落。人脸识别的优势是新颖的交互方法和人脸的不变性，劣势是复杂的环境下识别率低。研究的目的便是实现人脸识别，并且提高人脸识别的准确率，并且将其运用到实际生活中。

随着支付宝刷脸支付，FaceU检测人脸实时添加表情，腾讯QQ人脸登录等人脸识别技术在移动端的应用，人脸识别技术在移动端的应用也是越来越广泛。学习人脸识别技术在移动端的应用具有重要的意义。

## 1.3 研究内容及目标

本文主要的研究内容是人脸检测与人脸识别在安卓终端的应用，其中主要内容有：

(1)人脸检测算法涉及到Haar分类器的研究。

(2)人脸识别算法涉及到LBP人脸识别算法与样本训练。

(3)研究使用基于opencv的JavaCV库。

(4)研究安卓客户端开发技术,。

项目的研究目标有：

1. 实现80%的人脸检测成功率。
2. 实现在非复杂背景下50%的人脸识别成功率。
3. 实现安卓客户端录入人脸样本并且完成训练。

## 1.4 论文组织结构

本文一共分为7章，各章节具体介绍如下：

第一章：绪论。主要介绍了人脸识别的发展前景与历史以及当前人脸识别主要使用的技术，讲述了研究人脸识别的目的和意义，最后列出了研究的具体内容与目标。

第二章：相关技术及开发工具简介。介绍了搜索引擎的组成和RSS新闻源的特点，然后对此次系统的开发工具和开发技术做了大体的介绍。

第三章：新闻搜索引擎需求分析。对系统的每一个模块进行功能需求说明，对整个系统进行非功能需求说明。

第四章：系统数据库与详细设计。主要说明了系统的设计思路与功能结构，并对每一个模块进行说明。并且介绍了数据库的相关设计。

第五章：系统实现。首先介绍了代码整体结构，然后介绍了每一个子系统的具体实现流程，并附上核心代码与截图，同时对核心算法进行了分析与比较。

第六章：关键问题及解决方法。主要说明了系统开发中遇到的关键问题和处理方法。

第七章：总结与展望。对基于web的新闻搜索引擎系统的开发工作进行了总结，分析了有待改进与完善的部分。

# 第2章 相关技术及开发工具简介

## 2.1 搜索引擎

搜索引擎是指根据一定的策略、运用特定的计算机程序从互联网上搜集信息，再对信息进行组织和处理后，为用户提供检索服务，将用户检索相关的信息展示给用户的系统。搜索引擎技术源自于传统的全文检索技术，基本程序是将所有文章包含的词条抽出，建立以这些词条为关键词字的索引，然后通过用户的输入，在索引表中找到匹配的文档序列，再将结果返回给用户。一般搜索引擎主要由爬虫子系统、索引子系统和搜索引擎客户端组成。

爬虫子系统的主要任务就是对web上的站点以某种方式进行遍历，按照特定的遍历算法和选择机制对遍历到的节点进行下载。从遍历范围进行划分，爬虫可以被分为两类：一类是穷尽型爬虫，按照深度优先或者广度优先的策略，尽可能大而多的爬取整个web世界中的网页，这种爬虫一般应用于通用性商业搜索引擎，如百度、谷歌等专业的以搜索为主要业务的公司；第二类是选择爬取型爬虫，这种爬虫在爬取到了页面之后会对页面进行某种标准的分析评判，具体的标准取决于该搜索引擎的用途和领域，不符合要求的会被直接丢弃，因此，抓取页面的效率和分析效率会大大提高，由于网页特征集中，目标性更强，后期为这些页面提取词条，建立索引也更加容易快速，从而使得信息的有效性和时效性也远远高于前者。

索引子系统的主要任务就是在较短时间内，对爬虫子系统抓取到的网页进行处理，找到与用户输入关键词匹配的网页并给出反馈。建立索引的核心原理就是为每个原始网页文件建立对应的关键词索引记录，便于快速的根据用户的数据检索到该文件。为了给所有的文档进行相关度排序，还必须记录网页文件对应每个关键词的权值。大致步骤为先提取出原始网页文件的文本内容，然后利用分词系统对文本内容进行分词处理、关键词提取和权值计算，然后将结果存储到本地数据库或者文件系统中。建立索引的具体方法有很多，本系统采用的方法是在数据库中维护一张庞大的存储网页-关键词对应权值的索引表。

搜索引擎客户端的主要任务是为用户提供搜索接口，将符合用户要求的网页进行分级排序，然后将结果显示给用户。

## 2.2 RSS介绍

爬取子系统最大的难点就是如何从浩繁的web站点中找到符合要求的新闻页面，单纯的在web站点上进行深度优先或广度优先搜索效果均不理想，爬取的页面可利用率太低，筛选条件复杂。因此可以选择从各大站点的RSS源爬取新闻信息[8]。RSS的全称是Really Simple Syndication，它是基于文本的格式，是XML的一种形式。RSS快速而准确地沟通内容提供商和用户之间的联系，缩短了信息延迟。RSS文件通常只包含简单的项目列表，一般而言，每一个项目都含有一段标题和一段简单的介绍。RSS是站点之间共享内容的一种简易方式，它帮助用户在信息膨胀的web世界中获取对自己有价值的内容，从而避免了大量垃圾讯息。用户可以根据自己的喜好，在提供RSS订阅的站点订阅自己感兴趣的内容，客户端便可以获得这些内容的定时推送。同时用户可以看到没有广告和图片的标题或文章的概要阅读，这样不必阅读全文即可知道文章大意，为用户节省了时间。现在很多著名的新闻站点都提供了RSS订阅功能，定时抓取这些地址的RSS文件即可直接获取最新的新闻页面。

## 2.3 开发工具

（1）Sublime Text 3

sublime Text是一款极为出色的代码编辑器，它拥有非常漂亮的用户界面和强大的功能，它本身的架构非常简洁，但是强大的插件系统可以为用户提供几乎一切所需的扩展功能，包括而不仅限于：语法高亮、拼写检查、代码缩略图、仿控制台运行代码、多选择、多窗口、文件路径管理、即使项目切换、完整的Python API，支持几乎所有主流编程语言的语法，同时它还是跨平台编辑器，深受广大程序员的喜爱。仅仅几个简单的插件就可以将它从一个普通的文本编辑器搭建成一个强大的有完整开发能力的IDE。

（2）Chrome

Chrome是谷歌公司在2008年推出的一款浏览器。该浏览器是基于其他开源软件编写，包括WebKit,目标是提升稳定性、速度和安全性，并创造出简单且有效率的用户界面，是目前全球范围内使用最广的浏览器，对于普通用户而言，它兼容性强大，有丰富的扩展插件，速度快，安全性好；对开发者而言，它提供的开发者工具可以是前端工程师很方便的对代码进行调试，无论是审查元素还是查看网络数据传输都非常方便，是前端开发的debug利器。本次系统主要就是基于Chrome对搜索引擎系统进行开发和调试。

## 2.4 开发技术

（1）Python语言介绍

Python是一门面向对象、解释型的语言，因此它可以直接用解释器运行，无需编译。Python语法简洁而清晰，具有丰富而强大的类库，提供了所有脚本语言的简单和易用性，并且具有在编译语言中才能找到的高级软件工程工具。Python的基础类库覆盖了包括网络、文件、GUI、数据库、文本等大量内容，因此，使用python可以非常快速的构建出系统原型，而且其优雅、简单、明确的设计定位可以使我们在开发过程中将精力集中在算法和业务逻辑上，而避免了其他语言繁琐严格的语法和数据结构。目前Python有2.x和3.x两个版本，这两个版本并不兼容，在语法上有大量修改，虽然3.x版本解决了包括编码在内的大量问题，并且也会是以后的发展趋势，但是很多第三方库并没有完成从2.x到3.x的迁移，因此，本次系统开发选用的依然是兼容性最好的Python2.7[9]。

（2）FLASK框架介绍

web开发要经常和Http协议打交道，但是如果直接用Python处理http协议的话工作量会非常之大，除此之外，页面渲染、路由处理、session、cookie等web应用常常会遇到的问题也非常繁琐，因此，编写一个web应用常常需要一个框架来处理这些底层细节，开发人员可以将精力集中在业务逻辑的编写上。Flask框架是一个用Python编写的轻量级web应用框架，它提供小型的统一工具，或者在已有的API上构建的简单网络接口。其WSGI工具箱采用Werkzeug，模板引擎使用Jinja2，自带开发用服务器，集成了单元测试，基于Unicode并且有着丰富的文档可供参考，因此对于那些开发小项目、需要快速制作一个简单的Python支撑的网站的开发者很有用。

（3）Jieba中文分词库

中文分词就是把中文的汉字序列切分成有意义的词，中文分词的结果直接影响到关键词提取效果和相关度排序，是索引子系统的基础也是重要组成部分。中文分词技术属于自然语言处理技术范畴，现有的分词算法可分为基于字符串匹配的分词方法、基于理解的分词方法以及基于统计的分词方法。由于本次系统开发并不是研究自然语言处理，因此分词部分直接选用成熟的中文分词库。

Jieba中文分词库是目前python社区中最著名的一个中文分词库，它支持以下三种分词模式：

* 精确模式，试图将句子最精确地切开，适合文本分析；
* 全模式，把句子中所有的可以成词的词语都扫描出来, 速度非常快，但是不能解决歧义；
* 搜索引擎模式，在精确模式的基础上，对长词再次切分，提高召回率，适合用于搜索引擎分词。

并且该分词库提供了tf-idf，TextRank和基于前缀词典的词图扫描等算法实现，可以方便的调用接口提取关键词并得到对应的各类型权值。

# 第3章 新闻搜索引擎需求分析

## 3.1 功能需求

新闻搜索引擎分为爬取子系统、索引子系统和web客户端三个部分，系统的功能模块划分既要注重各个部分本身的功能，又要注重各个功能模块之间的联系，从而形成一套快捷高效的新闻搜索引擎系统。各个模块既拥有较强的独立性，又要留有适当的接口，以便日后对功能进行扩充和完善。

## 3.1.1 爬取子系统功能需求

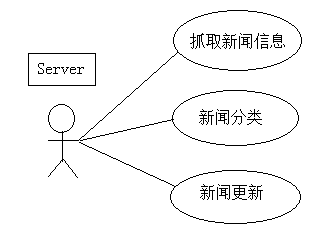


图3.1 爬取子系统功能需求图

搜索引擎具有广泛收集因特网上的Web页面，构建一个信息空间的作用。爬取子系统应实现的基本功能就是在各大新闻网站上抓取新闻信息，包括标题、链接、提要、正文等内容。为了突出新闻搜索引擎的独特性，区别于一般的搜索引擎，该部分在功能上还应做到以下几点：

（1）时效性。时效性是新闻的最大特点之一，因此本搜索引擎应实时更新，以保证抓取到各大新闻站点最新的新闻。

（2）新闻分类。用户搜索新闻往往带有一定的目的性，为了提高搜索精度，更精确的帮助用户找到想要的新闻条目，需要对所有新闻条目进行分类，包括：热点新闻，体育新闻，国际新闻，国内新闻，科技新闻，财经新闻，娱乐新闻。

## 3.1.2 索引子系统功能需求

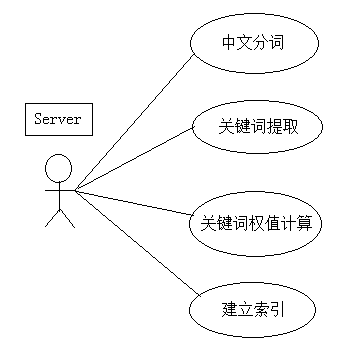


图3.2 索引子系统功能需求图

新闻搜索引擎的索引最基本的功能是为检索提供服务，将爬取到的新闻正文进行分词处理后，提取出权值较高的若干个关键词，在数据库中建立链接—关键词—权值的索引表。当用户输入检索词时，利用信息检索算法，尽可能将最相关的页面返还给用户，从而达到有效检索的目的。

随着抓取的新闻信息的增多、网页的不断变化，需要对数据库中的索引进行更新，部分网页不再存在，需要及时从索引库中删除。抓取到重复的新闻网页时，也应在数据库中对索引表进行去重处理。

## 3.1.3 web客户端功能需求

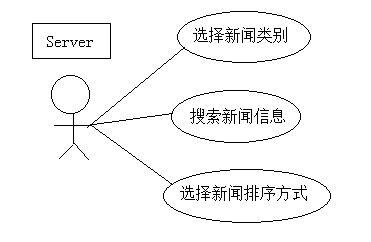


图3.3 web客户端功能需求图

web客户端给用户提供检索界面，并根据用户的查询要求，从数据库中检索出相关的信息并返回给用户。因为该界面直接面对最终用户，它是最终用户浏览、检索信息的入口，所以搜索界面应该符合用户习惯、操作简单、界面清爽。

操作上是用户在web客户端输入检索词，此时可以选择新闻类别，既可以大范围的在所有新闻中搜索，也可以在不同的新闻类别中进行针对性搜索。然后点击搜索按钮提交，服务器接受检索词并进行初步处理，将检索词拆分成多个子关键词，按照数据库中的索引表计算出符合用户查询情况的新闻链接及该条新闻的权值，然后按权值从大到小的顺序对相关新闻进行排列，最后根据结果渲染页面，返回给用户。

综上所述，这样一个基于web的新闻搜索引擎应，用户可以在浏览器上输入系统服务器的地址即可访问到系统的主页，然后输入检索词并且选择类别，服务器经过处理之后就会把一系列结果按照相关度从大到小的顺序返回给用户。用户根据标题、摘要、类别、发布时间来判断最符合自己要求的新闻条目，然后点击链接即可访问原始新闻网页。

## 3.2 非功能需求

为了更好地用户体验，该搜索引擎系统也需要以下几点非功能需求：

（1）可用性：新闻客户端应该有良好的UI设计，用户能够方便的操作使用。搜索出的每一条结果应该清晰简明的展示出标题、链接、简述、发布时间等信息。

（2）可靠性：该搜索引擎和存储新闻的数据库的设计应该能够承受较大的访问量而不崩溃。

（3）健壮性：对于用户的不合法输入有一定的处理能力，对于找不到搜索结果的输入也能够给出提示。

（4）安全性：新闻搜索入口可以防止恶意的SQL注入攻击。这是一种常见的攻击方式，有的系统会直接运行用户的输入，对后台数据库进行攻击，所以要做相关的处理。

（5）可维护性：搜索引擎系统的设计应该考虑到后期的维护升级，爬虫子系统、索引子系统和客户端之间相互独立，做到高内聚，低耦合，方便后期对某一部分进行维护升级时，不影响其它部分的功能

# 第4章 系统数据库与详细设计

## 4.1 数据库设计

良好的数据库结构对于保障一个应用程序始终在高性能状态具有非常重要的意义。数据库如果没有设计好将会对以后的工作带来很多麻烦，一方面是性能问题，另一方面是维护问题。过多的重复性数据会严重影响系统的执行性能。

## 4.1.1 实体属性图

实体属性图反映的是一种数据关系，实体是现实中存在的对象，可以是具体的事物，也可以是抽象的概念。本文中的新闻就是一个实体，在本系统中，它具有若干特性，例如，每一条新闻都有其唯一的标题、链接等内容，这些特征就称之为实体的属性。新闻的实体属性图如图4.1：

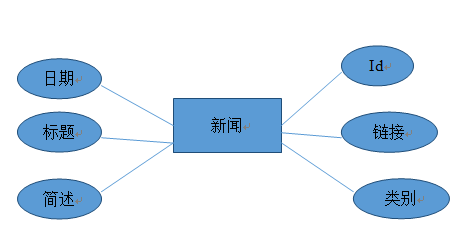


图4.1 实体属性图

## 4.1.2数据库设计

根据本系统的实际需求，数据库并不复杂，仅包含两张基本表，一张表存储新闻的基本信息，另一张表存储索引。

（1）新闻表：用来保存新闻的所有关键信息，包括新闻的日期、标题、简述、id、链接、类别。（如表4.1所示）

表4.1 新闻表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否主键 | 默认值 | 描述 |
| pubDate | date | 否 |  | 新闻日期 |
| title | varchar | 否 |  | 新闻标题 |
| summary | varchar | 否 |  | 新闻简述 |
| id | Int | 是 | 自增 | 新闻编号 |
| link | varchar | 否 |  | 新闻链接 |
| category | varchar | 是 |  | 新闻类别 |

（2）索引表：用来保存每条新闻链接的正文中关键词的权值。（如表4.2所示）

表4.2 索引表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否主键 | 默认值 | 描述 |
| link | varchar | 否 |  | 新闻链接 |
| keyword | varchar | 否 |  | 关键词 |
| wvalue | float | 否 |  | 关键词的权值 |

## 4.2 系统详细设计

整个新闻搜索引擎系统主要包括三部分的功能模块：爬虫子系统、索引子系统和web客户端（如图4.2所示）。这三个部分分别负责：管理rss源并爬取新闻网页存入数据库，对数据库的每条新闻进行分词处理、关键词提取、建立索引、处理用户输入及相关度排序，处理用户交互。每一个子模块之间功能相互独立，互不影响功能完整性。

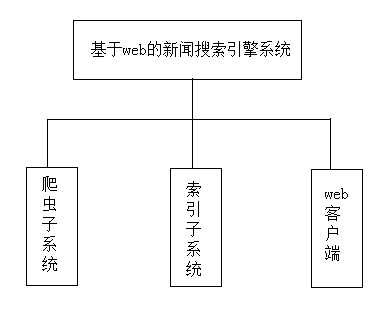


图4.2 系统功能结构图

## 4.2.1 爬虫子系统模块设计

爬虫子系统主要由RSS源控制、页面解析以及新闻信息存储三个部分组成，其结构模块如图4.3所示：

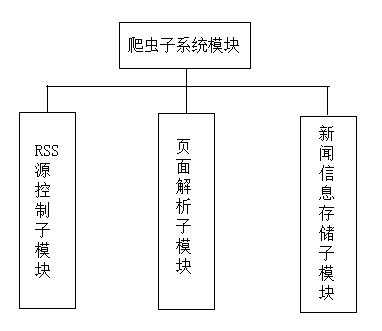


图4.3 爬虫子系统功能模块结构图

设计思路：一般的搜索引擎爬虫都会从源网页开始，以深度或广度优先的方式对整个站点进行遍历，由于本系统的搜索主题是新闻，采用这种类型的爬虫速度慢，获取的页面可利用率极低，服务器的压力也较大，后期对页面的分析处理工作也极为繁琐。因此，本系统采用的是直接爬取腾讯、网易的RSS新闻订阅源，这样的好处在于目标明确，不仅可以保证爬取的全部都是最新的新闻，并且在爬取阶段即可确定新闻的类别，由于RSS是XML的形式，因此解析起来也相当轻松，可以直接从XML文件的项目节点获取每条新闻的标题、简述、日期、链接等，这大大简化了后续的分析工作。

爬虫子系统的功能流程如下：从所有目标新闻站点的RSS订阅地址抓取XML文件 ，遍历XML文件，解析XML文件中所有的新闻条目，将所有新闻条目经过索引子系统的处理后存入数据库，每间隔一段时间重复一次以上过程。

## 4.2.2 索引子系统模块设计

索引子系统主要由新闻页面解析、关键词提取及权值计算、建立索引及存储以及相关度排序这四个部分组成，其结构模块如图4.4所示：

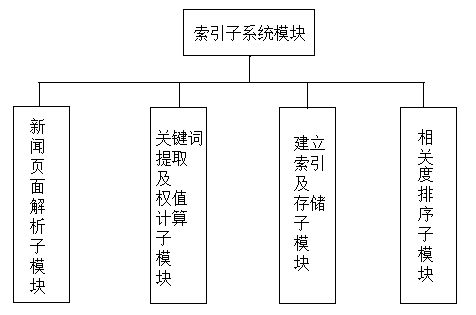


图4.4 索引子系统功能模块结构图

设计思路：索引子系统的目标是根据新闻正文、标题等内容，在数据库中维护一张庞大的索引表，记录每个关键词对应的链接及权值。因此索引子系统首先应该得到新闻的正文及标题，然后利用jieba进行关键词提取技术得到每条新闻对应的关键词，最后根据TF-IDF算法计算出每个关键词的权值，这就完成了一次索引建立。

索引子系统的功能流程如下：将所有爬虫爬取的新闻条目进行以下处理：分析处理新闻页面正文部分，得到每一条新闻的标题，类别，发布时间，概要，链接。然后利用分词器对正文进行处理，得到关键词，再根据相关算法计算每个关键词对应的权值，最后将所有的信息存储到数据库中。当用户输入检索词时，系统会先将检索词拆分成若干个关键词，然后根据这若干关键词对应权值表计算出每条相关新闻的总权值，按权值从大到小的顺序将新闻结果进行排序，从而返回给用户。

## 4.2.3 web客户端模块设计

Web客户端主要由搜索入口页面、结果页面、服务器端这三个部分组成，其结构模块如图4.5所示：

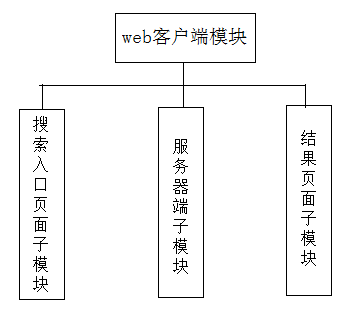


图4.5 web客户端功能模块结构图

web客户端的功能流程如下：用户访问服务器地址时，返回到浏览器系统主页，用户在搜索入口页面输入关键词请求搜索服务时，先对用户的输入进行分词处理，然后根据相关算法从数据库中的索引表中找到所有记录并进行权值计算，按照从大到小进行排列，最后将这些信息返回给web服务器，web服务器生成结果页面发送给用户，用户点击每条搜索结果可得到详细的新闻内容页面。

# 第5章 系统实现

## 5.1 系统总体结构

系统总体结构如图5.1

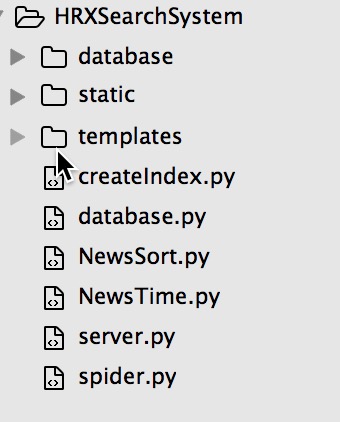


图5.1 总体结构图

本系统使用的是flask框架，文件结构和通用web应用类似。database文件中存储的是和数据库操作有关的文件。

static文件夹中存储的是静态文件，包括CSS，fonts，images，JavaScript等文件。

template中存储的是页面模板，即待jinja2模板引擎渲染的html页面。

后面的py文件是服务器端运行的python文件，server.py是服务器主进程，负责监听端口，响应客户端请求，处理用户输入，从数据库索引表中得到新闻条目计算权值并且渲染页面等。

Spider.py 是爬虫子系统的主要代码，负责将新闻爬取下来并且调用createIndex.py中的方法为新闻建立索引记录，将相关信息存入数据库中。

NewsSort.py、NewsTime.py、datebase.py分别处理有关相关度排序，新闻事件处理，数据库操作相关的一些业务逻辑代码，将这部分重用性较高的代码分离出来有利于降低系统内代码的耦合度，便于后期对这部分代码进行修改和维护。

## 5.2 爬虫子系统实现流程

首先在腾讯和网易的RSS新闻订阅源上，按照体育、国际、国内、财经、科技和娱乐等类别收集RSS源并放在一个字典中。feedparser是较为通用的用于解析rss格式的python模块，它的强大不仅是因为它对各种rss版本的兼容性好，而且使用起来非常简单，只用feedparser的parse函数就可以方便的对rss源进行下载并解析，d = feedparser.parse(url)，这样rss的所有内容就都在d里面了。用feedparser遍历所有的rss源，对每一个rss源的所有新闻项目进行分析，获取到日期、标题、提要、链接和类别这些信息之后，将这些信息一条一条地存入数据库中。

网络爬虫从RSS新闻源提取新闻相关信息并存入数据库的核心代码如下：

#将所有rss源分类放在一个字典中

rssUrls = {'体育':['http://sports.qq.com/basket/rss\_basket.xml',

'http://sports.qq.com/isocce/rss\_isocce.xml',

'http://sports.163.com/special/00051K7F/rss\_sportsyc.xml'],

'国际':['http://news.qq.com/newsgj/rss\_newswj.xml'],

'国内':['http://news.qq.com/newsgn/rss\_newsgn.xml',

'http://news.163.com/special/00011K6L/rss\_sh.xml'],

'财经':['http://finance.qq.com/financenews/domestic/rss\_domestic.xml'],

'科技':['http://tech.qq.com/web/webnews/rss\_11.xml',

'http://tech.163.com/special/000944OI/hulianwang.xml'],

'娱乐':['http://ent.163.com/special/00031K7Q/rss\_entmovie.xml']}

#提取每个rss源的每条新闻的类别,日期,标题,提要,链接

def getItemsInfo(category,rssUrl):

itemList = []

d = feedparser.parse(rssUrl)

items = d.entries

for item in items:

date = item.published

if "GMT" in date:

date = NewsTime.wyNewsTimeFormat(date)

title = item.title

summary = item.summary

link = item.link

#为每条link在数据库建立索引

# createIndex.createIndex(link,title+summary.split('<')[0])

# createIndex.createIndex(link,title)

itemList.append((category,date,title,summary,link))

return itemList

#爬虫主进程

def spider(rssUrls):

for category,urls in rssUrls.items():

for url in urls:

itemList = getItemsInfo(category,url)

#把抓取的内容存入数据库

database.saveToDB(itemList)

该部分运行截图如图5.2:

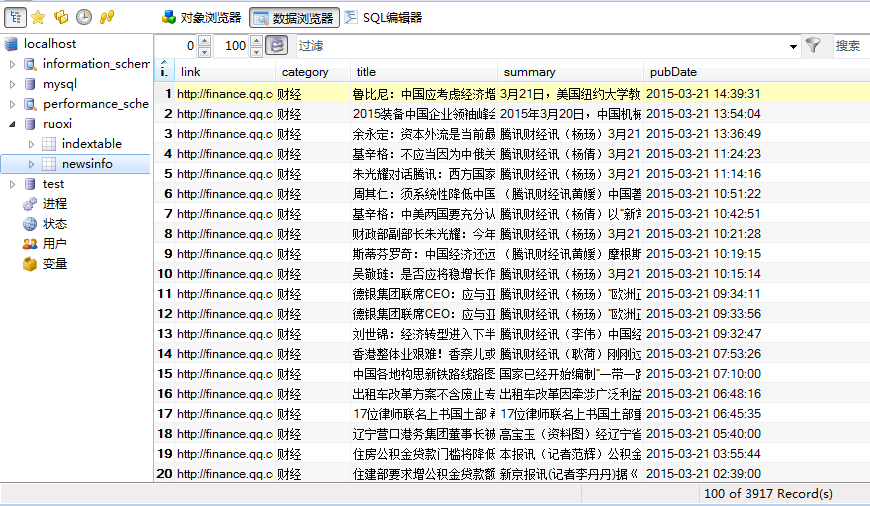


图5.2 爬取子系统截图

## 5.3 索引子系统实现流程

## 5.3.1 关键词提取算法分析

关键词提取就是从文本里面把与这篇文章意义最相关的一些词抽取出来。这个可以追溯到文献检索初期，当时还不支持全文搜索的时候，关键词就可以作为搜索这篇论文的词语。因此，目前依然可以在论文中看到关键词这一项。除此之外，关键词在文本聚类、分类、摘要等领域中也有着重要的作用。比如从某天所有新闻中提取出这些新闻的关键词，就可以大致知道那天发生了什么事情。

关键词提取从方法来说大致有两种：一种叫做关键词分配，就是给定一个关键词库，然后对于一篇文章，从词库里面找到几个词语作为这篇文章的关键词。另一种是关键词抽取，就是对于一篇文章，从文章中抽取一些词语作为这篇文章的关键词。目前大多数领域无关的关键词抽取算法（领域无关算法的意思就是无论什么主题或者领域的文本都可以抽关键词的算法）和它对应的库都是基于后者的。从逻辑上说，后者比前者在实际使用中更有意义。另外，从结果的角度来说，关键词抽取也可以分为两种。一种是仅仅把词语抽取出来，这个相对简单，容易实现，比如FudanNLP、jieba、SnowNLP。另外一种则是连词和短语一起抽取出来，这个还需要增加短语抽取这一个步骤，这一类的实现包括ICTCLAS、ansj\_seg等。常见的关键词抽取算法有TextRank算法和TF-IDF算法。

关键词特征之一就是在文本中反复出现且关键词附近出现关键词的概率非常大，因此就有了TextRank算法[10]。它基于PageRank算法，将文本中每个词看成一个页面，认为文本中某一个词语与之周围N个词存在一个link，然后在这个网络中使用PageRank算出每个词语的权值，把权值最高的几个词作为关键词即可。TextRank典型的实现包括FudanNLP和SnowNLP等。

使用TextRank提取关键字，首先将原文本拆分为句子，然后在每个句子中过滤掉停用词，并只保留指定词性的单词，由此可以得到句子的集合和单词的集合。每个单词作为pagerank中的一个节点。设定窗口大小为k，假设一个句子依次由下面单词组成：，，，，，…，。其中，，…，、，，…，、，，…，等都是一个窗口，在一个窗口中的任两个单词对应的节点之间存在一个无向无权的边。基于上面构成图，可以计算出每个单词节点的重要性，最重要的若干个单词就可以视作关键词。

最为常见且最易实现的关键词提取算法就是TF-IDF算法，它可以找到文本中常见但是在别的文本中不常出现的词语，这正好符合关键词的特点。字词的重要性随着它在文件中出现的次数成正比增加，但同时也会随着它在索引库中出现的频率成反比下降。TextRank实际应用效果并不比TF-IDF有明显优势，而且由于涉及网络构建和随机游走的迭代算法，效率极低。TF-IDF具有较强的普适性，该算法基本能应付大部分关键词抽取的场景。使用这种方法典型的例子是jieba。

本系统采用的便是TF-IDF算法，其算法原理如下：

1.先给本聚类内所有文档进行分词，然后用一个字典保存每个词出现的次数，即词频(term frequency, TF)。由于同一个词语在长文件里可能会比在短文件有更高的词频，而不管该词语重要与否，所以该数字通常会被归一化，以防止它偏向长文件。

2.遍历每个词，得到每个词在文档里的逆向文件频率 (inverse document frequency, IDF)，即在所有文章里出现的倒文档频率。某一特定词语的IDF值，可以由总文件数目除以包含该词语的文件的数目，再将得到的商取对数得到。

3.用一个key为词，value为TF\*IDF权重的字典来保存所有词信息，然后按value对字典排序，取权重高的若干个词作为关键词。

对于在某一特定文件里的词语来说，它的重要性可表示为：

其中，是该词在文件中出现的次数，分母是文件中所有字词的出现次数之和。

其中，是语料库中的文件总数，||是包含词语的文件数目。如果该词语不在语料库中，就会导致被除数为零，因此一般情况下使用1+||。

然后

某一特定文件内的高词语频率，以及该词语在整个文件集合中的低文件频率，可以产生出高权重的TF-IDF。因此，TF-IDF倾向于过滤掉常见的词语，保留重要的词语。

## 5.3.2 搜索引擎排序算法分析

排序算法往往是搜索引擎的核心竞争力之一，用户行为研究显示，通常用户只会关注搜索结果的前几条，越往后面的结果越少有人关注。因此改进排序算法，让用户最感兴趣最关注的内容排在前面是提升用户满意度的关键。常用的搜索引擎排序算法有：PageRank算法、HITS算法、Direct Hit算法等[11][12]。

PageRank 算法是当前最为主流的搜索引擎排序算法之一，它是一种和查询词无关的算法，在用户查询之前进行离线计算网页等级，不占用查询时的计算时间。现已成功运用于Google搜索引擎。PageRank 算法的中心思想是一个网页的重要性决定并且依赖着其他网页的重要性，一个网页被许多其它网页引用，则它可能是重要页面；一个网页尽管没有被多次引用，但如果它被一个重要的网页引用，则它也可能是重要页面；一个网页的重要性被平均的传递到它所引用的网页。

PageRank算法的优点在于它对互联网上的网页给出了一个全局的重要性排序，并且算法的计算过程是离线完成的，在用户查询时仅仅根据关键字匹配获得网页集合，然后排序推荐给用户，这样有利于迅速响应用户的请求。但该算法还存在不少缺陷：主题无关性，无法区分页面内的导航链接、广告链接和功能链接等，容易对广告页面有过高评价，从而导致出现“主题漂移”问题。另外，旧的页面等级会比新页面等级高，因为新网页通常拥有很少的链入网页数量，所以其PageRank值会大大低于其他网页。对于时间因素比较敏感的页面，如本文讨论的新闻页面，实时性是一个重要相关因素，因此不适合运用此算法。

HITS算法也是利用被链接网页的质量和数量来确定搜索结果的权重值。该算法将网页分为两种类型：一种是表达某一主题的权威页面，称为Authority页面，依赖于指向它的页面；另一种是能把Authority页面联结在一起的页面，称为Hub页面，依赖于它指向的页面。由于网页的这种分类，HITS算法也就涉及了两个重要的权值，Authority值和Hub值。Authority表示一个权威网页被其它网页所引用的加权数量，即该权威网页的加权入度值，若某网页被引用的数量越大，则该网页的加权入度值越大，Authority越大；Hub表示一个Web页面指向其它网页的加权数量，即该Web页面的加权出度值，它提供了指向权威页面的链接集合，某网页的加权出度值越大，则该网页的Hub值越大。通常情况下，好的Hub页面指向很多好的Authority页面；好的Authority页面也会有许多Hub页面所指向。HITS算法正是利用了页面间的这种关系，从而大大提高了搜索质量。

HITS算法的优点是它能更好地描述互联网的组织特点，由于它只是对互联网中很小的一个子集进行分析，所以它需要的迭代次数更少，收敛速度更快，减少了时间复杂度。但HITS算法也存在一些问题：第一，中心网页之间的相互引用以增加网页评价，某些网页可能通过恶意大量引用其它页面，来提高自身排名；第二，主题偏离，由于该算法只是根据链接关系确定权重，不会确认页面是否有效，忽略了页面链接之间的差异性，把所有的链接都看作具有同等的重要性，当查询很多广义主题时，HITS算法会把一些和搜索主题无关的链接给予很高的价值度，尤其当这些链接自身也是中心网页或权威网页时。例如，当用户输入查询关键词“电影人物”时，一些电影公司的网站主页很有可能会排在最前面，因为电影公司网站商业性存在的原因，许多链接会在这些公司网站的主页之间产生，从而导致“主题偏离”现象。

PageRank算法和HITS算法都是客观的描述了网页之间的本质特征，但它们都很少考虑到用户浏览习惯时的主题相关性。与这些算法相比，Direct Hit算法是一种重视信息质量和用户反馈的排序方法。它的基本思想是，系统将查询的结果提交给用户，接着开始跟踪用户在检索结果中的点击行为，如果网页被用户点开进行浏览，并且浏览的时间较长，则可认为该网页的受欢迎程度就高，相应地，系统将增加该网页的相关度。反之，若用户点开网页后停留时间较短，并且重新返回点击其他网页，那么认为该网页与查询词具有很小的相关性，系统将降低该网页的相关度。由此可见，采用这种算法，相关度在不断变化，对于同一个词在不同的时间进行检索，得到的排序结果可能不尽相同，网页的排名取决于被点击的次数和被浏览的时间长度，用户点击反馈的多少决定了算法的精确度，所以说Direct Hit算法是一种取决于用户检索行为的动态排序算法。

Direct Hit算法的优点是直接融入用户的反馈信息，考虑了用户的主观因素，能够保证页面的质量。然而缺点是，该算法只适用于检索关键词较少的情况，因为它实际上并没有进行排序，而是一种筛选和抽取。用户行为具有很强的随意性，在检索数据库很大、关键词很多的时候，返回的结果有很多页，用户不可能一条条查阅，多数用户只会关注前几页显示的内容，因此一些排名较后的网站获得的点击率就会很低，这样很难提升网页的排名。所以，Direct Hit算法不能作为主要的排序算法来使用，它是一种很好的辅助排序算法，目前在许多搜索引擎中仍然在使用。

分析与比较了多种相关度排序算法后，本系统采用的相关度排序方法是使用TF-IDF模型[13][14]。TF-IDF是一种统计方法，用以评估单词对于一个文件集或一个语料库中的其中一份文件的重要程度。TF-IDF模型的主要思想是：如果词w在一篇文档d中出现的频率高，并且在其他文档中很少出现，则认为词w具有很好的区分能力，适合用来把文章d和其他文章区分开来。信息检索时，对于每个文档，都可以分别计算一组搜索词的TF-IDF，将它们相加，就可以得到整个文档的TF-IDF。这个值最高的文档就是与搜索词最相关的文档。

TF-IDF算法的优点是简单快速，结果比较符合实际情况。缺点是单纯以“词频”衡量一个词的重要性，不够全面，有时重要的词可能在文章中出现的次数并不多。此外，这种算法无法体现词的位置信息，出现位置靠前的词与出现位置靠后的词都被视为重要性相同，这是不合理的。

## 5.3.3 建立索引的实现流程

Jieba中文分词是python的中文分词组件，它不仅具有分词功能，还具有添加自定义词典、关键词提取、词性标注等多种功能。首先利用jieba的关键词提取功能对每条新闻的正文进行关键词提取并计算其TF-IDF值，keywordsList = jieba.analyse.extract\_tags(content,withWeight = True)，content为新闻正文，即待提取的文本，withWeight为是否一并返回关键词权重值，将其设置为True。然后将每条链接—关键词—权值记录存储到数据库的索引表中。在用户输入搜索词后，先利用jieba分词功能的搜索引擎模式对用户输入的检索词进行分词处理，keywordslist = jieba.cut\_for\_search(inputSentence)，分词后得到n个关键词，然后从数据库的索引表中得到这n个关键词对应的结果序列，将这些序列中链接相同的记录的权值进行相加，最后将所有记录按总权值从大到小进行排列，即作为最终返回结果的顺序。

由一组关键词列表得到排好序的新闻条目列表的核心代码如下：

#根据keywordsList返回sorted linkslist

def getSortedList(keywordsList):

linkDict={}

for each in keywordsList:

print each

cursor.execute('''select link,wValue from indexTable where keyword=%s''',[each])

values = cursor.fetchall()

for each in values:

link = each[0]

wValue = each[1]

if link in linkDict.keys():

print "hi"

linkDict[link] = (linkDict[link] + wValue)\*100

print link

else:

linkDict[link] = wValue

sortedTuplelist = sorted(linkDict.items(),key=lambda a:a[1])

newlist = [i[0] for i in sortedTuplelist]

return newlist[::-1]

#派生HTMLParser基类,解析html，提取关键词

class myHtmlParser(HTMLParser.HTMLParser):

url = ''

newsId = ''

def handle\_starttag(self,tag,attr):

if tag == 'meta':

if ('name','keywords') in attr:

keywords = attr[1][1]

titleKeywordsList = jieba.analyse.extract\_tags(keywords,withWeight = True)

finalList = titleKeywordsList

if len(finalList) is not 0:

database.saveToIndexDB(self.url,finalList)

该部分运行截图如图5.3

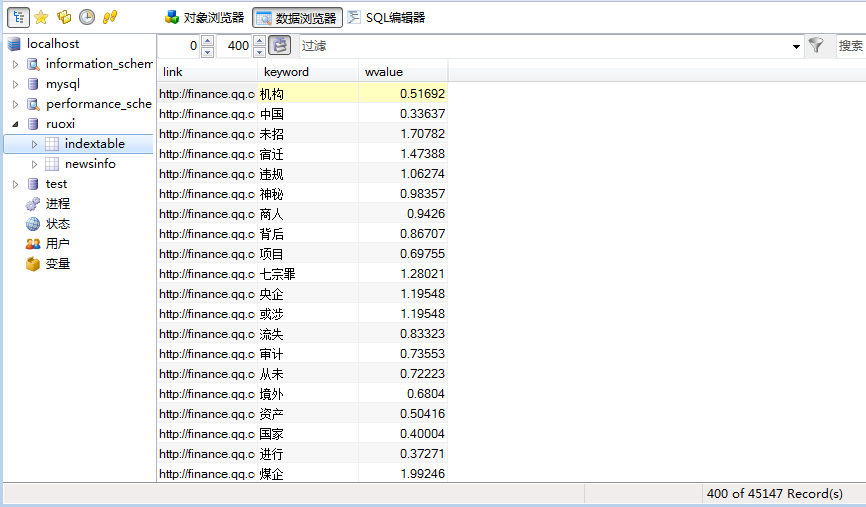


图5.3 索引子系统截图

## 5.4 搜索引擎框架搭建

## 5.4.1 B/S架构设计

本次客户端开发采用的是典型的B/S架构，服务器主要为客户端提供两个页面，一个是搜索框页面，用户在这个页面进行输入操作，还有一个是结果页面，服务器将搜索结果显示在这个页面。服务器要实现的核心业务逻辑仅包括根据用户输入的关键词来返回对应结果。

## 5.4.2 UI设计



图5.4 UI设计图

## 5.4.3 前端框架开发流程

开发客户端时，首先用easy\_install工具通过命令行安装好了flask框架，在Sublime Text中编写好服务器主程序后，测试运行了服务器端代码，在浏览器中，访问http://127.0.0.1:5000 成功访问到了服务器提供的“Hello Flask”页面。接下来编写系统主页面，即搜索入口，用户访问该地址时，服务器端将主页面返回给客户端。根据设计好的主页面，用html、css编写好主页面，将css文件、Jquery库等静态文件分别放入系统根目录的/static/css和/static/js文件夹下，此时访问该地址，便可以成功显示出主页。接下来要能够使服务器端接收到用户的请求参数，包括搜索关键词，搜索种类和排序方式。通过form表单形式向服务器端发送带参数get请求，然后在服务器端主程序中，编写好处理该请求的路由方法/search，根据参数得到对应的结果列表，然后根据结果列表，动态生成结果页面，生成结果页面的方式是通过Jinja2模板引擎，将结果列表传递给Jinja2模板引擎之后，Jinja2通过自身的循环控制代码及html代码，动态生成每一项结果，包括新闻标题、日期、简述和链接等。最后在Sublime Text中运行服务器主程序，即可通过浏览器访问该地址，使用到该系统的检索服务。

运行截图如图5.5



图5.5 程序运行截图

# 第6章 关键问题及解决方法

## 6.1 爬取子系统部分

为了突出新闻搜索引擎的特点，在爬取新闻信息时加入了新闻分类这个属性，用户在使用该系统时，可以先选择自己感兴趣的新闻类别，然后在所选类别下进行进一步的查找，这样可以缩小查询范围，节省查询时间，提高查询结果的准确性。

由于选取的新闻源来自腾讯和网易两个网站，因此爬取到的新闻时间格式不一致。腾讯的时间格式形如2015-01-01 13:52:20，网易的时间格式形如Thu，19Mar 2015 01:12:11 GMT，网易采用的是格林威治时间。通过上网查询到解决方法，首先要将格林威治时间转换成北京时间，先用切片选取网易时间GMT以外的部分，然后转换成时间戳，再转换成datetime格式，这样时间上就可以进行加减，让小时部分加8小时，从而转换成北京时间。最后统一格式，先用time.strptime()将时间的数字部分以元组形式存放，然后用time.strftime()函数将时间统一改成纯数字的形式，这样方便之后对新闻的时鲜度进行比较。

该模块进行测试时，发现爬虫有时候耗时较长，长时间无响应。这是因为某些网站自身原因或网速不稳定，对某些新闻链接的访问请求耗时较长，使程序长时间阻塞。为了防止这类链接破坏爬虫的工作效率，设置socket的超时时间为10s，若10s内未得到响应，则抛弃该链接，继续请求下一个链接。这样保证了爬虫的工作效率和稳定性。

## 6.2 索引子系统部分

在对系统进行测试时，输入单关键词和输入多关键词时检索到的结果几乎一致。打印权值之后发现，该关键词的权值异常的高。调试后发现这是因为索引表中的某些记录有大量重复。由于当前的索引子系统每次都会在爬虫爬取了RSS新闻后马上为它建立索引，然而很多RSS源上的新闻长期不更新，因此定期爬取时可能会多次爬取到该新闻，以至于多次为该新闻建立了索引，计算权值时也计算了多次，导致再对这些词进行检索时，第二个关键词的权值几乎不起作用。

对数据库进行去重操作后，再修改了索引部分的代码，确定爬取的新闻链接成功存入数据库时（表明之前未爬取过），然后再对新闻建立索引。问题解决后，输入多关键词时，检索的结果表现出很高的相关度。

## 6.3 web客户端部分

最开始开发前端页面时，将所有的html、css文件都放在了根目录下，通过使用Chrome浏览器的开发者工具调试后发现，服务器并没有将css文件发送给浏览器，查阅资料后了解到，flask框架使用的Jinja2模板引擎对文件路径有特殊规定，它会默认按照该规定进行路径搜索，静态文件必须放入根目录的static文件夹下，而js，images，css等静态必须放入文件指定的文件夹下才能被正确获取并发送给客户端。按如上所述建立了文件目录后，重新把所有文件放入指定文件夹内并修改了代码内相关路径后，问题成功解决。

在服务端运行程序时有时出现socket.error: [Errno 48] Address already in use的错误。这种错误的原因是因为出现了端口占用，因为flask运行时一直在监听5000端口，而上次程序没关闭时再次运行就会端口被占用，程序无法运行的状况，在命令行中通过“lsof -i tcp:5000”的命令找到占用5000端口的进程pid，然后“kill pid”杀死该进程，即可正常运行现有的服务器程序。

# 第7章 总结与展望

基于互联网和信息技术的发展，网络信息检索与分享具有重要影响，搜索引擎受到越来越多用户的关注，因此研究和实现一个搜索系统显得很有意义。本文研究了基于Web的新闻搜索技术，并在此基础上实现了一个Web新闻搜索平台。现对本文所做工作做个总结：

本文首先对搜索引擎的概念和分类进行介绍，特别是在新闻方面做了进一步了解。在系统开发时，查阅资料后采用抓取RSS新闻源的方法，从网上爬取新闻信息。然后在关键词提取技术上分析了TextRank算法和TF-IDF算法，同时分析与比较了多种排序算法，并联系当前互联网现状浅谈了这几种算法的优点和所面临的不足。最后对本次新闻搜索引擎系统的开发流程进行了详细的介绍。

由于时间和精力有限，本论文在对搜索引擎系统的研究中，只是对它的基本理论和基本框架进行了学习，所开发的新闻搜索引擎系统虽然具有新闻搜索的基本功能，但还有很大的改进空间：

1.没有实现索引与数据库数据自动更新功能。自动更新对每日都要进行数据更新的新闻搜索引擎来说是必不可少的，要在后期阶段进行实现。

2.对于用户个性化信息的获取方面，除了使用用户输入检索词外，还可以通过用户对页面的停留时间和链接的收藏等情况来获取用户兴趣偏好。

3.新闻分类方面，可将TF-IDF值和余弦相似度一同使用于向量空间模型中，以此判断两份文件之间的相似性。当然，除了使用文本分类技术进行分类外，还可以通过聚类技术对用户进行聚类，这样可以优化用户兴趣所属类别的划分结果。

4.相关度排序方面，TF-IDF算法中并没有体现出单词的位置信息，对于Web文档而言，权重的计算方法应该体现出HTML的结构特征。特征词在不同的标记符中对文章内容的反映程度不同，其权重的计算方法也应不同。因此应该对于处于网页不同位置的特征词赋予不同的系数，然后乘以特征词的词频，以提高文本表示的效果。

5.新闻的相关度排序很大程度上和用户平时的搜索行为与搜索爱好相关，因此，如果能考虑到用户的兴趣偏好与行为习惯，将能够更好的为用户展示出最匹配的内容。

总之，互联网的高速发展在给广大用户提供便利的同时，也给信息搜索领域带来了很大的挑战。为了满足用户日益变化的要求，相关领域的技术也需要不断改进与创新。搜索引擎系统的真正内涵是非常丰富的，还有许多问题需要解决，许多方法需要论证。深入研究和探讨相关内容也是我以后需要继续努力去做的工作。

# 参考文献

[1]朱健琛.个性化新闻搜索引擎的研究与设计[D].成都：成都电子科技大学，2012.

[2]张玉宝.中文搜索引擎的原理及相关技术分析[J].福建电脑，2008(1)：7.

[3]张文静.搜索引擎的分类及发展趋势[J].焦作大学学报，2006（3）: 76- 78.

[4]常晓燕.基于Java的新闻搜索引搜索的设计与实现[D]. 成都：西南交通大学，2004.

[5]王益明，刘菲.中文搜索引擎搜索结果重合率研究报告[R].2007.

<http://www.searchlab.com.cn/thesis.php>.

[6]赵敏涯.基于主题的新闻搜索引擎的研究与实现[D].扬州：扬州大学，2006.

[7]林子熠.基于模板的Web新闻搜索技术的研究与发现[D].上海：上海交通大学，2009.

[8]钱爱兵.基于RSS的Web新闻主题聚合系统的设计与实现[J].现代图书情报技术，2007(4).

[9]Martin C.Brown. Python：The Complete Reference[M]. McGraw-Hill Company，2001.

[10]Mihalcea R, Tarau P. TextRank: Bringing order into texts[C]//Proceedings of EMNLP. 2004,4(4): 275.

[11]文军舰.基于Nutch的Web结构挖掘算法研究[D].天津：天津财经大学，2011.

[12]黄言之.浅谈搜索引擎的核心算法[Z].2013.<http://www.williamlong.info/archives/3526.html>.

[13]图灵社区.TF-IDF模型的概率解释[Z].2012.<http://www.ituring.com.cn/article/15381>.

[14]博客频道.tf-idf详解.2013[Z].<http://blog.csdn.net/july_2/article/details/9698755>.

[15]王启户.面向远程教育的搜索引擎系统设计与实现[D].西安：西安电子科技大学，2008.

[16]尹浩.基于WWW的新闻搜索引擎的设计与实现[D].成都：西南交通大学，2003.

[17]王晶.基于Web信息获取的新闻数据分析研究[D].上海：华东师范大学，2009.

[18]徐宝文，张卫丰.搜索引擎与信息获取技术[M].第1版.北京：清华大学出版社，2003.

[19]胡小睿.基于Web挖掘的搜索引擎技术研究[D].武汉：武汉大学，2005.

[20]林伟业.基于爬虫的Sohu新闻搜素引擎设计与实现[D].广州：中山大学，2012.

[21]曾小芹.基于领域本体的新闻搜索引擎的研究与实现[D].南昌：南昌大学，2012.

[22]Christos Bouras，Vassilis Poulopoupos，Panagiotis Silintziris. Personalized News Search in WWW：Adapting on user’s behavior. [Fourth International Conference on](http://61.183.148.149:8000/rewriter/IEEE/http/hdddwoknqd9hddd9nqf/xpl/mostRecentIssue.jsp?punumber=5072479) Internet and Web Applications and Services[Z],2009.

# 致谢

时光飞逝，大学四年的学习生涯即将结束，首先我要向我的母校——武汉理工大学致以诚挚的谢意。在这里，我度过了人生中一段美好而又难忘的时光。值此论文完成之际，我要感谢我的导师岑丽副教授，论文从选题、撰写、修改到定稿，系统从开发、调试、修改到优化，每一步都得到了岑老师的精心指导，这对我毕业设计的顺利完成起到了重要的作用。同时，谨向所有关心我学业的老师、同学、朋友表示衷心的感谢。

本论文从开题到完成经历了一个学期，看着这厚厚的文稿，轻松之余，更多的是一种由衷的感动。完成毕业设计过程中，岑老师对我的论文提出了许多宝贵的建议，我遇上难题向她询问时，她也耐心解答。岑老师渊博的知识，严谨的治学态度，优良的工作作风，谦逊的人格给我留下了深刻的印象，这会是我一生学习的典范。除了学业的悉心教导和严厉鞭策外，生活上岑老师也经常关心我们。有幸成为岑老师的一名学生，是我一生受用不尽的财富。

我要感谢计算机学院的领导和老师们。作为软件工程的学生，他们传授的宝贵知识是使我论文得以顺利完成的坚实基础，我们学院丰富的教学资源开拓了我的眼界。这一切都是我今后从事专业工作的强力后盾。然后，我要感谢四年里和我朝夕相处的同学们，我们从陌生到相识相知，一起上课、一起学习讨论、一起生活，这是一种缘分。他们给予的真挚的友谊、真心的关爱、真诚的鼓励、真切的微笑也会伴随我的一生。

特别要感谢我的父母，感谢他们对我从小到大的关爱与教诲，对我学习工作的支持与鼓励，对我无知幼稚行为的宽容，感谢他们一直以来的无私奉献与不求回报。在学业完成之际，向含辛茹苦的父母表示由衷的谢意和崇高的敬意。

面对如此之多的关心与帮助，常感无可回报，唯有在未来的工作和生活中时时敦促自己，取得更加丰富的成果，才不负亲人、师长和朋友的殷殷期望之情和谆谆教诲之意！