硕士学位论文



题目： 基于网页信息自动提取的  
分布式爬虫系统设计与实现

学 号： 201814022()

姓 名： 杨本栋

专 业： 电子与通信工程

导 师： 夏海轮

学 院：信息与通信工程学院

中国-北京

&玄彳倉*様*

硕士学位论文



题目：基于网页信息自动提取的  
分布式爬虫系统设计与实现

学 号： 2018140220

姓 名： 杨本栋

专 业：电子与通信工程

*导* 师： 夏海轮

学 院：信息与通信工程学院

Beuing University of  
Posts and  
Telecommunications

Thesis for Master Degree



Topic: The Design and ImDlementation of

Distributed Web Crawler Svstem

Based on Automatic Extraction of

Webpa—e Information

|  |  |
| --- | --- |
| Student No.: | 2018140220 |
| Candidate: | Bendong Yang |
| Subject: | Electronic and |
|  | Communication Engineering |
| Supervisor: | Hailun Xia |
| Institute: | School of Information and |

Communication Engineering

26th May 2021

独创性（或创新性）声明

本人声明所呈交的论文是本人在导师指导下进行的研究工作及取得的研充 成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不 包含其他人己经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他 教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任 何贡献均已在论文中作了明确的说明并表示了谢意。

申请学位论文与倍料若有不实之处，本人承担一切相关责任。

本人签名： 日期：A|.-牛

关于论文使用授权的说明

本人完全了解并同意北京邮电大学有关保留、使用学位论文的规定，即：北 京邮电大学拥有以下关于学位论文的无偿使用权，具体包括：学校有权保留并向 国家有关部门或机构送交学位论文，有权允许学位论文被查阅和借阅；学校可以 公布学位论文的全部或部分内容，有权允许釆用影印、缩印或其它复制手段保存、 汇编学位论文，将学位论文的全部或部分内容编入有关数据库进行检索。（保密 的学位论文在解密后遵守此规定）

本人签名:

导师签名:

日期：AW. *牛*

日期：*加*, 6心

基于网页信息自动提取的分布式爬虫系统设计与实现

摘要

数据爆发式增长，大数据时代到来，互联网每天产生难以计数且 各式各样的数据。这些数据蕴含的信息具有巨大的研究价值和商业价 值。学者及企业希望获取知识类、资讯类、政策法规类等文章型网页 中有价值的信息，不仅要求数据量大、格式规范统一、时效性强，还 希望获取信息的成本低、效率高。现实中网页结构各异且都包含大量 与主题无关的网页噪声，如何从互联网蕴含的海量信息中以较高的效 率、极快的速度获取有价值的结构化信息是一个值得研究的课题。

本文选题来源于企业项目，研究文章型网页的信息自动提取算法, 设计实现了基于网页信息自动提取算法的分布式网络爬虫系统，提高 了信息获取的效率，具体如下：

1. 针对现有网页信息提取算法提取精度不足、提取信息缺失 和上下文信息利用较少的问题，提出了一种基于视觉块一致性和序列 标注的文章型网页元数据提取算法。根据文章型网页的视觉特征对网 页进行分块预处理，将网页节点划分为多个一致性视觉块；利用统计 特征定位网页主体区域，过滤掉大量噪音信息；选择文本、视觉和词 典特征作为特征集进行特征提取，利用条件随机场模型进行序列标注, 提取标题、正文、作者、来源、发布时间、图像和附件等信息。最后 对算法进行了实验和对比分析。
2. 设计实现了基于网页信息提取算法的分布式爬虫系统。本 文分析了企业需求并对系统进行了总体设计，将系统分为数据釆集层、 数据解析层、数据存储层、节点接入层和系统管理层。针对现有分布 式爬虫存在的问题，引入网页信息自动提取算法代替人工编写解析脚 本，提出了一种无中心的基于动态反馈的任务调度策略，提高了系统 可靠性和抓取效率。接下来本文对系统各模块进行了设计与实现。最 后，本文对系统进行了性能和功能测试。

关键词：网页信息提取自动化分布式爬虫

THE DESIGN AND IMPLEMENTATION OF DISTRIBUTED  
WEB CRAWLER SYSTEM BASED ON AUTOMATIC  
EXTRACTION OF WEBPAGE INFORMATION

ABSTRACT

With the explosive growth of data volume and the arrival of big data era, the Internet generates countless and variable data every day. The information contained in these data has great research value and commercial value. Scholars and companies hope to obtain the latest and valuable information from article-type web pages such as news web pages, policies and regulations web pages. Not only do they require large amounts of latest data in a consistent format, but they also hope to obtain information with low cost and high efficiency. In reality, web pages have different structures and contain a large amount of noise that has nothing to do with 比e subject. It is a problem that how to obtain valuable structured information from the massive information contained in the Internet at a relatively fast rate and efficiency. And the problem is worth studying.

This thesis is selected from the enterprise project. The research proposes an article-type web page information extraction algorithm based on visual block and nodes sequence annotation, and designs and implements a distributed web crawler system based on this algorithm, as follows:

(1) Aiming at the problems that existing web page information extraction algorithms are not so accurate, lack information items and cannot make full use of contextual information. This thesis proposes an article-type web page metadata extraction algorithm based on visual block consistency and sequence annotation. Firstly, according to the visual features of the aiticle-type webpages, this thesis preprocesses the webpages into blocks, divides the webpage nodes into multiple consistent visual blocks; Secondly, this thesis locates the main area of the web page and filters out a lot of noise information by using statistical features; Next, this thesis selects text features, visual features and dictionary features as feature set, and uses conditional random field model to annotate sequences to extract information such as title, text, author, source, release time, images and attachments. Finally, the thesis verifies the effect of the algorithm through experimentation and comparison, and the result shows that the algorithm has a better extraction effect.

(2) This thesis designs and implements a distributed crawler system. This thesis analyzes the needs of the enterprise, and designs the overall architecture of the system and each layer of the system. The system is divided into data acquisition layer, data analysis layer, data storage layer, node access layer and system management layer. Aiming at the problems of existing distributed crawlers, this thesis uses automatic extraction algorithm for web page infoimation instead of manually writing parsing scripts and proposes a centreless task scheduling strategy based on dynamic feedback, which improves the system reliability and efficiency. This thesis describes the design and implementation of the system in sub-modules. Finally, this thesis testes the perfbnnance and fiinctionality of the system.

KEY WORDS: web page information extraction automatic distributed crawler

目录

[摘要 I](#bookmark16)

ABSTRACT II

[第一章绪论 1](#bookmark27)

1.1研究背景及意义 1

1.2主要研究内容 2

1.3论文结构安排 3

[第二章相关技术综述 5](#bookmark32)

2.1网络爬虫技术 5

[2.2常见网页结构分类 8](#bookmark48)

[2.3网页信息提取技术 10](#bookmark61)

2.3.1基于模板的网页信息提取 11

2.3.2基于启发式规则的网页信息提取 11

2.3.3基于视觉分块的网页信息提取 13

2.3.4基于机器学习的网页信息提取 15

[2.4分布式系统技术概述 16](#bookmark84)

2.4.1分布式数据存储 16

2.4.2分布式任务调度 18

2.4.3面向服务架构 19

[2.5本章小结 19](#bookmark87)

[第三章文章型网页信息自动提取算法研究 21](#bookmark90)

[3.1引言 21](#bookmark93)

[3.2文章型网页分析 22](#bookmark96)

[3.3基于视觉块一致性和序列标注的文章型网页元数据提取算法 23](#bookmark105)

[3.3.1网页预处理 24](#bookmark108)

3.3.2网页分块与网页主体区域定位 25

3.3.3基于CRF的网页元数据抽取 30

[3.4实验分析 33](#bookmark160)

3.4.1数据集及评价标准 33

3.4.2实验结果与分析 34

[3.5本章小结 36](#bookmark163)

[第四章分布式爬虫系统的设计与实现 37](#bookmark166)

[4.1引言 37](#bookmark169)

[4.2系统需求分析 37](#bookmark172)

[4.2.1功能需求 38](#bookmark175)

4.2.2非功能性需求 38

[4.3系统总体设计 39](#bookmark184)

[4.4系统模块设计与实现 41](#bookmark192)

4.4.1存储模块的设计与实现 41

4.4.2网页下载与任务调度模块的设计与实现 44

4.4.3网页信息提取模块的设计与实现 51

4.4.4网页去重模块的设计与实现 54

4.4.5节点管理模块的设计与实现 58

4.4.6爬虫管理模块的设计与实现 62

[4.5系统测试 64](#bookmark238)

[4.5.1功能测试 64](#bookmark241)

[4.5.2性能测试 65](#bookmark244)

[4.6本章小结 67](#bookmark247)

[第五章总结与展望 69](#bookmark250)

[5.1论文工作总结 69](#bookmark253)

[5.2下一步工作展望 70](#bookmark259)

参考文献 71

缩略语 75

[致谢 77](#bookmark326)

第一章绪论

1.1硏究背景及意义

2020年9月，由国家互联网络信息中心牵头并组织进行第46次互联网发展 现状的调查M 截止到2020年6月为止，我国的互联网公民己达到9.40亿，互 联网的覆盖人数更是达到全国人口的67%； 2020年1至6月，移动互联网接入 流量达745亿GB,同比增长34.5%,网页数量在2018年底就己达到2816亿个。 随着互联网规模的不断扩大，互联网中的信息数量以指数级速度增长，互联网俨 然己经成为了最大的数据库。互联网蕴含的信息具有巨大的研究价值和商业价值, 比如新闻网站上的新闻数据，对新闻进行聚类将其分为多个话题，追踪分析不同 话题的趋势以及倾向性，最终给决策者或监管者提供可靠的分析数据目。通常网 页中都包含大量用于网站推广或商业宣传的与主题无关的内容，同时海量的网页 中充斥着大量重复冗余的网页信息数据，约占30%-45%团。海量的数据同样使我 们面临着信息过载和信息冗余的问题，如何从互联网蕴含的海量信息中高效的获 取有价值的信息是当前的研究热点之一。相比于人工釆集数据的方式，信息抽取 技术和爬虫技术可以帮助我们高效地釆集大规模数据。

网页数量巨大，随之而来的是各种各样的网页结构，针对每一个网页人工编 写解析脚本是一个繁琐的任务回，为每个单独的网站开发爬虫代码的复杂性成为 严重限制研究机构和企业快速获取数据的瓶颈。同时一个网站的网页结构也不是 一成不变的，这使得维护爬虫的成本十分高昂，如何应对网页结构的变化也十分 重要。信息抽取能够自动地从非结构化文档中识别和提取结构化信息，能够从海 量数据中快速准确的分析岀真正有用的信息，可以提高信息采集的效率。在网页 信息抽取领域，研究人员对不同类型的网页提出了各式各样的网页信息抽取算法。 随着网页设计规范和网页设计风格的变化，有的算法己经不再适用。因此，研究 实用、高效的网页信息抽取算法是Web信息检索领域面临的一个新的挑战［5〕。 网页信息自动提取算法的深入研究具有十分重要的价值。

面对互联网中海量的数据以及专家学者对信息获取时效性的要求，单机式爬 虫己经不能满足信息抓取的需要，分布式爬虫的出现解决了这一问题。目前已有 部分分布式爬虫系统，面向搜索引擎的分布式爬虫系统过于复杂，应用性不高〔4】； 部分单机式爬虫框架通过改进可以实现分布式网页爬取，但由于框架的约束性， 导致使用起来并不灵活。面对以指数级速度的增长的互联网数据，如何设计实现 高效率、高健壮性的通用爬虫软件，仍是现在的研究热点之一。

综上所述，基于网页信息自动提取的分布式爬虫系统的提岀，有效解决了网 络爬虫面对海量数据抓取性能低的问题，提高了信息获取效率，本文具有以下意 义：

进入大数据时代以及各式网页的出现，传统的单机网络爬虫很大程度上已经 不能满足当前的抓取需求〔6】，而人们对信息获取实时性和准确性的要求越来越髙。 高效的网页信息提取算法和分布式网络爬虫系统的提出非常必要，本文研究提出 新的网页信息提取算法，并将此算法引入分布式网络爬虫系统，这对于解决当前 信息获取效率低下的问题具有非常大的意义。

本文依托于与中国电信集团系统集成有限责任公司合作的项目，主要负责采 集国内外教育信息，为企业某信息平台提供数据，对解决企业问题具有一定的现 实意义。

1.2主要研究内容

本文提出了一种新的文章型网页信息自动提取算法，设计实现了分布式爬虫 系统。本文的主要研究内容包含以下两个方面：

1. 文章型网页信息自动提取算法

针对现有文章型网页信息提取算法提取精度不足、提取信息缺失和上下文信 息利用较少的问题，提出了一种基于视觉块一致性和序列标注的文章型网页元数 据提取算法。算法能够自动提取新闻、博客文章和其他文本密集的文章型网页中 的元数据，如标题、正文、作者、日期信息、图像和附件等信息，将非结构化或 半结构化数据转化为结构化数据。首先分析文章类型网页的通用视觉特征和统计 特征，在分析特征时不考虑节点类型，根据节点的视觉特征和统计特征对节点进 行分块，将所有节点分为多个一致性视觉块，计算每个视觉块的特征选出网页主 体区域，使用条件随机场(Conditional Random Field, CRF)对网页主体区域的视 觉块进行标注，识别网页元数据，完成网页信息的自动提取。最后通过实验验证 算法的效果并与现有提取方法进行了对比，结果表明该算法具有较高的准确率。

1. 分布式爬虫系统的设计与实现

第二部分设计实现了基于网页信息自动提取算法的分布式爬虫系统。系统能 够协调分布于网络各位置的节点协同完成网页下载和网页信息提取任务,然后进 行网页去重和信息存储。本文分析了企业需求，然后对系统架构进行设计，针对 现有分布式爬虫存在的问题，引入网页信息自动提取算法代替人工编写解析脚本,

提出了一种无中心的基于动态反馈的任务调度策略，提高了系统可靠性和抓取效 率。接下来分模块对系统进行了实现，最后对系统进行了功能测试和性能测试。

1.3论文结构安排

本文基于机器学习技术和企业需求研究了网页信息自动提取算法并设计实 现了基于此算法的分布式爬虫系统。论文结构安排如下。

第一章，绪论。主要介绍本课题的研究背景和意义，概述了本课题的研究内 容并给出本文的结构安排。

第二章，相关技术研究综述。首先介绍了网络爬虫的基础性知识和技术，主 要包括爬虫基本结构、网络爬虫分类和分布式爬虫的网络结构等方面，对比了主 从式结构和自治式结构的优缺点。接下来介绍了网页结构分类和网页信息自动提 取算法。最后介绍了分布式系统的相关技术。

第三章，网页信息自动提取算法研究。深入分析了文章型网页的特点，针对 现有网页信息提取算法提取精度不足、提取信息缺失和上下文信息利用较少的问 题，提出了一种基于视觉块一致性和序列标注的文章型网页数据提取算法，给出 了关键环节的算法描述。最后对算法的效果进行了验证。

第四章，基于网页信息自动提取的分布式爬虫系统设计与实现。首先结合企 业实际需要进行了需求分析，接下来对系统逬行总体设计，从下至上的将系统分 为数据釆集层、数据解析层、数据存储层、节点接入层和系统管理层，然后对系 统各层中功能模块进行了设计和实现，最后从功能测试和性能测试两方面对系统 进行测试。

第五章，总结与展望。本文对所研究的信息提取算法和分布式爬虫系统进行 了总结，通过分析论文中的不足寻找解决方案，对下一步工作进行展望。

北京邮电大学工程硕士学位论文

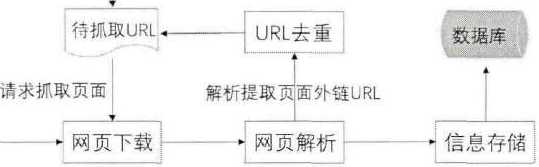
第二章相关技术综述

基于网页信息自动提取的分布式爬虫系统的研究与实现涉及的理论与技术 主要与网络爬虫技术、网页信息提取技术和分布式系统相关技术有关。本章对以 上技术的相关研究做了介绍。

2.1网络爬虫技术

网络爬虫(Crawler)是指一种从互联网上爬取信息的程序或者脚本。其结 构如图2-1所示。

种子URL



图**2-1**通用网络爬虫的基本结构

种子统一资源定位符(Uniform Resource Locator, URL)集合，是爬虫运行 所需的入口地址，以一定的抓取策略开始爬取数据。

待抓取URL队列，该部分来自于种子URL和己抓取网页中解析出的经过去 重的URL。

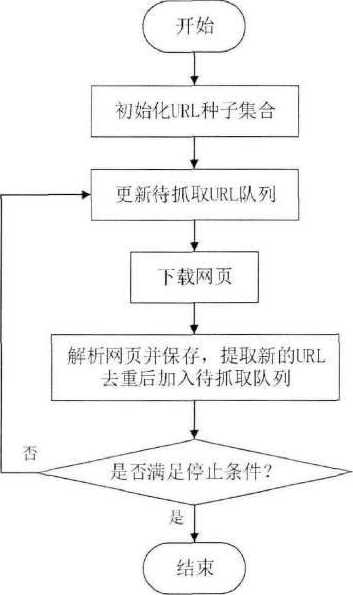
网页下载，使用超文本传输协议(HyperText Transfer Protocol, HTTP)协议 从互联网上下载URL对应的网页到本地。为防止网络波动等问题造成的网页下 载失败，通常需要设定超时重试机制，重试一定次数后放弃抓取访问时间过长的 网页，提高网页抓取效率。

网页解析，提取网页中的有效信息，同时提取网页中链接的其他页面的URL。

URL去重，同一个网页可能被多个其他网页链接到，为了避免重复抓取对 资源造成浪费，需要对新解析出来的URL进行去重处理，然后将去重后的URL 添加到待抓取URL队列也

信息存储，用于存储釆集到的网页URL、页面内容和解析后的结构化数据 等信息。

从通用型网络爬虫的基本结构可以看出，其工作流程可以概括为搜寻、下载 与解析，搜寻即在互联网上寻找目标页面，下载即模拟用户行为下载网页到本地, 解析即使用某种方式将网页中的非结构化或半结构化数据转化为结构化数据，同 时提取网页中链接的其他页面的URLo



图**2-2**爬虫通用工作流程图

爬虫的通用工作流程如图2-2所示，具体描述如下：

1. 根据抓取需求，选取合适的URL作为入口地址。
2. 将种子URL移入待抓取URL队列，并按预定策略对URL进行排序。
3. 从待抓取URL队列取出URL,访问并下载对应网页。
4. 解析网页，将网页与解析数据保存，从网页中解析出新的URL链接， 将经过去重后的URL加入待抓取队列。
5. 循环以上操作，直到满足抓取需求或抓取深度与抓取广度达到预设值。 网络爬虫通常可以分为通用网络爬虫、主题网络爬虫和增量网络爬虫同，实

际场景中使用的爬虫通常为几种爬虫的组合。

通用网络爬虫，又称全网爬虫，主要用于搜索引擎的网页数据釆集，是指从 种子URL开始采集数据，逐渐扩展到整个互联网的网络爬虫。因为互联网网站 众多、网页数量巨大，通用网络爬虫需要抓取的范围广、数量多，通常釆用分布 式架构，通过增加节点数量来提高抓取速度和存储效率。

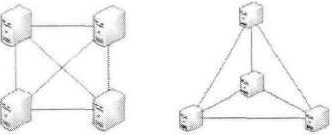
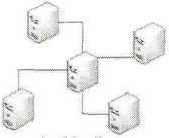
主题网络爬虫，是指根据特定算法按照预先设定的主题抓取与主题相关页面的网络爬虫⑼。与通用网络爬虫不同，它不是为了抓取整个互联网的网页，而是 专门用于对某个主题的网页进行数据釆集，有效的主题判定算法是其核心，现有 的主题判定算法基于以下三种方式：基于网页链接结构、基于网页文本内容和基 于混合的文本内容与链接结构。主题爬虫在捜索过程中过滤掉了很多无关信息， 使得爬虫系统更为高效。

增量网络爬虫，是指对目标网站进行增量式下载更新的网络爬虫3】。网页是 不断变化的，是不定时更新的，增量爬虫只能在一定程度上保证抓取的网页是尽 可能最新的。通常的做法是通过新鲜度与年龄指标来标记一个页面⑴］,通过周期 性抓取的方式来实现增量抓取，设置合适的时间间隔使得网页整体的新鲜度最高 或者年龄最低。增量网纟各爬虫的抓取过程十分复杂，实现起来有一定难度。

互联网时代数据量爆炸式增长，每时每刻都会产生新的数据，使用爬虫技术 可以即时抓取有效数据信息，将最新的数据用于各种分析研究。因此爬虫技术随 着互联网的发展而不断进步，传统的单机式爬虫已经逐步演进为分布式爬虫。

单机爬虫虽然可以通过多线程的方式去下载解析网页，但是由于单台计算机 资源的限制导致网页爬取的速率有限，如网络带宽限制和计算能力限制。解决这 一问题的直接方法是增大网络带宽和采用运算能力更强的计算机，然而这种方法 不能从根本上解决问题而且成本太高。解决这一问题最好的办法就是采用分 布式系统，将单机爬虫看作是分布式系统中的一个节点，多个节点同时运行，付 出的代价是多个节点间协同工作的通信和调度开销。本文所研究的就是此类分布 式爬虫。

分布式网络爬虫在由多个网络爬虫节点组成的集群上运行，其核心问题是管 理和协调各个节点间的工作。多个节点间协调工作的关键在于解决节点间的任务 调度问题。



主从式 自治式 混合式

困2-3分布式网络爬虫的拓扑结构困

任务调度需要节点间进行通信，根据节点间的通信方式，分布式网络爬虫的 拓扑结构可以分为主从式、自治式和混合式，如图2-3所示。

(1)主从式

由一个中心节点以及若干个爬虫节点组成。主节点控制协调整个集群的运行, 维护和管理待抓取URL列表，给节点分配URL抓取任务，从节点只需要运行主

节点分发的任务，下载并解析网页，提取新的URL返回给主节点。主从式结构 实现简单，易于进行抓取任务分配和节点监控管理。但是主从式结构存在主节点 性能瓶颈和单点故障问题，主节点的性能瓶颈同样会制约集群的规模，而且主节 点出现故障会导致整个系统的失效。

1. 自治式

与主从式不同，自治式没有一个专门的节点做协调工作，所有的节点是对等 的，通过节点间互相通信协调工作。每个节点都要维护、管理待抓取URL列表, 同时承担任务调度，将不属于自己的任务发送给其他节点。任务的划分通常采用 一致性哈希算法。自治式结构没有主节点，不存在单点故障问题和主节点性能瓶 颈，具有更好的健壮性和扩展性。但是节点间会频繁通信，这会产生很大的负担, 导致系统性能下降，而且节点无法感知集群整体的状态，可能会导致负载不均衡 问题。

1. 混合式

混合分布式结构结合上面两种模式的特点，含有一个中心节点和若干个爬虫 节点，中心节点不需要进行任务调度，只负责监控各节点的状态。

Nutch是一个开源的分布式爬虫框架，底层基于Hadoop实现，有着爬取速 度快的优点，可以进行累积式抓取和增量式抓取，但定制能力比较弱。马联帅冋 使用Redis替换Scrapy框架的待爬取队列实现主从结构的分布式爬虫。荣啥⑵ 将Scrapy与分布式流计算框架Storm结合实现分布式爬虫平台，以批处理的方 式进行URL调度。

网络爬虫在抓取网页时也要满足合法性和合理性。相比于人工访问网页，网 结爬虫访问下载网页的速度更快，这会对网站造成很大的负载，可能会影响正常 访问，网站运营者可能会釆取反爬虫措施封禁该爬虫，比如封禁爬虫IP。网络 爬虫在抓取数据时要遵守爬虫协议，同时要设置合理的抓取间隔以免对网站造成 过大的负载。

2.2常见网页结构分类

对目前互联网上常见的网页，根据表现形式和结构特征可以分为以下几种类 型：

(1)主题型网页

主题型网页主要指以文本主题内容为主的文章型网页，还包含以某种事物 为主题的网页，如产品页面等。



® 2-4典型文章型网页

图2-4是一个典型的文章型网页，文章型网页有如下特点：

1） 网页主体通常位于页面中央；

2） 网页主体由多个连续的文本段组成，包含大量文字和文本符号；

3） 文章标题非常显著，相较于其他文本字体非常大；

4） 大多数正文文本是左对齐且宽度相等的；

5） 大多数正文文本使用同一样式和大小的字体；

6） 作者、来源、发布时间等信息通常位于正文文本前后；

7） 部分网页的meta信息包含有标题和发布时间等信息。

（2）列表型网页

如图2-5所示，列表型网页列表中的每一项都具有相同的结构，通常通过链 接指向主题型网页。



图**2-5**列表型网页

(3)评论型网页

如图2-6所示，评论型网页由主题帖和用户评论发言组成，具有以下特点:

1. 只有一个用户发言区，由若干个用户评论区组成，位于网页中央;
2. 用户的每一个评论区都是相邻的，具有相同的宽度；
3. 用户评论区通常含有子评论区，每个评论区的内部结构相同；





图**2-6**评论型网页

2.3网页信息提取技术

网页信息提取，就是去除网页中的噪音数据，提取网页中有价值的信息。传 统的信息提取方式是通过人工编写解析表达式，通常有三种方法：正则表达式匹 配、XML 路径语言(XML Path Language, Xpath)匹配和样式选择器(CSS Selector)

匹配。人工编写解析表达式需要对抓取的每个网站编写相应的解析脚本，工作量 大，而且网站的网页可能会发生变化，解析脚本的维护也是难以解决的问题。随 着互联网的快速发展，网页信息自动提取技术成为研究的热点之一。

来自不同领域的研究人员对不同类别的网页提出了各式各样的网页信息自 动抽取算法，根据提取原理的不同，可以将算法分为基于网页模板的、基于视觉 分块的、基于启发式规则的和基于机器学习的。

2.3.1基于模板的网页信息提取

某个网站的页面或者某个领域的网页具有相似的网页布局，具有十分相近的 或者相同的结构，有些是基于同一个模板生成。基于模板的网页信息提取就是利 用这个特点，把网页中重复出现的内容看作模板，对网页归类总结生成模板，网 页信息提取时可以根据网站的特点匹配模板库中的相应模板获得所需要信息。根 据提取模板的使用的网页范围，算法可以分为两类，一类是从相同页面结构的网 页集中提取模板，一般从同一网站的不同网页中提取；另一类是归类各种不同的 网页，对每类网页分别进行提取，提取抽象层次更高、归纳性更强的通用模板卩叽 第一类方法提取的准确率非常高，显而易见的是通用性比较差，不适用于其他网 站的信息提取，第二类方法与之相反，通用性强但准确率低，某些场景下可以在 通用性和准确性方面取得很好的平衡，如商品列表页面、电影评论页面等。当网 页模板较多时可以建立模板库，通过模板匹配可以在一定程度上解决通用性差的 问题。模板匹配通常依赖于目标网站的入口地址、网站类型、网站的抓取状态和 网页DOM树的相似程度。

为了简化复杂模板的生成，嵌入浏览器的可视化模板生成器提供了易操作、 可视化的工具，通过在浏览器上简单的点击操作和选择操作，即可完成复杂的模 板编辑任务，如集搜客等工具。

Wang等人卩41提出DSE算法,其核心思想是对同一个网站中的网页构建DOM 树，对使用相同模板的网页进行DOM树匹配，把匹配结果中重叠的部分看作噪 声，把非重叠部分看作网页主体，提取非重叠部分叶子节点的内容oBar-Yossef等 人卩句釆用pagelet对DOM树进行分割以及检测模板。

基于模板匹配的方法具有准确性较高的有点，但是局限性大。针对某个网站 提取的模板一般不适用于其他网站，网站一旦进行更新或者改版就需要对模板进 行更新。

2.3.2基于启发式规则的网页信息提取

基于启发式规则的网页信息提取是对网页进行分析，从中归纳总结出目标内

容和其他内容的区别，根据区别制定相应的启发式规则，利用这些规则过滤网页 噪音，提取有效信息卩虬目前基于启发式规则的算法主要针对包含大量的文本信 息的网页，主要分为基于统计信息的和基于行块分布函数的正文提取算法。

基于统计理论的信息提取方式的依据是网页的正文部分文字密集，而其他区 域链接比较多。这类算法中比较典型的是基于文本链接比的提取算法，最后根据 文本密度或文本链接比是否达到阈值来判断是否为正文部分，在判断正文开始和 结束时，需要考虑上下文的文本密度或文本链接比以提高判断的准确度。该方法 适用于包含大量文本信息的页面，如新闻页面、博客等。但不同网页的阈值有所 差别，阈值设定的是否合理决定了信息提取效果。Gupta等人通过计算链接文 本长度与普通文本长度的比值，将其与预设阈值比较来定位到网页中的目标区域。 预设的阈值是关键，正文区域定位的准确性取决于阈值设置的合理性，一般通过 大量的试验和人工校验来确定。孙承杰等人泌〕提出了一种依靠统计信息提取网页 正文的方法，韩忠明等人卩％根据中文新闻与博客网页的正文特征利用超文本标 记语言(HyperText Markup Language, HTML)与文本的密度比进行文本的识别 与抽軌 MSong等人网提出了一种内容提取方法，该方法可以根据中文标点符 号的数量和非超链接字符数与超链接包含的字符数的比率来发现网页内容。王琦 等人［尚将HTML文档转换为STU-DOM树模型，根据非链接文本和链接文本的 统计特征，对其进行节点过滤和剪枝。王少康等人提出使用特征文本密度来提 取正文的方法，能够较为准确地识别复杂网页以及多主题段网页的正文信息。但 是总体来看，仅仅使用基于统计理论的信息提取方式对大部分文本密集的网页提 取效果不佳。

基于行块分布的正文提取算法的依据是网页正文部分连续且文本较多［制。该 算法首先过滤网页源码，只留下文字文本部分，去除网页标签等其他文本，按行 将文本分割，挑选出符合正文特性的行块作为正文。通常正文区域是连续的，只 需找到行块的起始位置和终止位置，一般为文本分布的骤升骤降点，基于行块分 布的抽取过程的伪代码如表2-1所示。

基于行块分布的算法较基于统计理论的算法的优点在于计算简单，不需要分 析源码构建DOM树，但是过于依赖正文在网页源码中的位置分布，正文中掺杂 了其他信息或正文分布于网页源码的几行中都会造成提取错误。

基于启发式规则的方法能获得较高的准确率，但启发式规则是人工大量实验 确定的，只适用于某一领域的网页，而且启发式规则是基于当前网页建立的，网 页变化可能会导致启发式规则不再适用。

第二童相关技术综述

表**2-1**基于行•块分布的抽取算法

算法：基于行块分布的抽取算法

输入：网页文档，正文长度的阈值threshold,行块厚度k 输出：正文文本

**ExtractContent:**

1按行分割文档，总行数记为len,所有文本行组成数组lines

1. boolstart=boolend=false
2. WHILE i小于等于len-k
3. 第i行至第i+k行组成一个行块，存入行块数组block
4. 计算block(i)的长度存入数组blockjen
5. END WHILE
6. WHILE j小于数组blockjen的长度
7. If block\_len(j)大于 threshold 并且 boolstart=false
8. 记录正文特征行开始位置starts
9. boolstart=true
10. If boolstart=true并且block\_len(j)等于0并且其后k个值都等于0
11. 记录正文特征行结束位置end-i
12. boolend=true
13. END If
14. END If
15. END WHILE
16. 提取start与end 之间的正文内容

2.3.3基于视觉分块的网页信息提取

基于视觉分块的网页信息提取是根据页面的视觉信息EJ,网页源码经浏览器 渲染之后，从用户的视觉角度，根据区块大小和位置以及空白区域可以轻松地将 网页划分为不同的视觉块，视觉块的内部结构和内容不会影响对一个独立的视觉 块的判断，视觉块内部的一些视觉信息，如字体的样式、视觉快的位置和大小、 文本中段落间的距离等信息也可以充当判断独立视觉块的条件。

该方式最经典的算法是微软研究院提出的VIPS(Vision-based Page Segmentation)算法岡，其流程如图2-7所示。目前很多相关的研究都是在VIPS 算法上进行的。

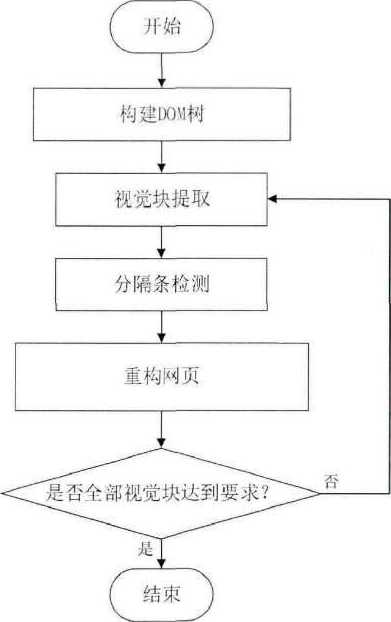


图2-7 VIPS算法流程图

VIPS中提出使用反映视觉块内容联系紧密程度的内聚度DoC(Degree of Coherence)用来划分语义一致的网页块。DoC具有以下特点：

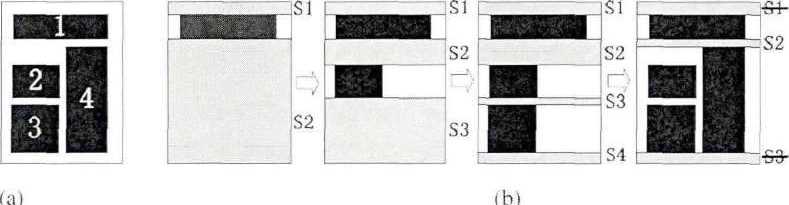
1. DoC值越大，视觉块内的内容越一致。
2. 在层次结构树中，父级的DoC大于儿子节点的DoC。

在视觉块提取过程中，根据VIPS算法的判定规则计算岀DoC值，DoC值 的取值范围是［1,10］。在进行页面分块前，还需要预先设定一个用于限定视觉块 分割粒度的PDoC值，通过适当的PDoC将网页细分为不同的粒度。PDoC越小， 内容结构就越粗糙。

在完成第一轮视觉块检测之后，通常一个网页会被分割为几个视觉块，按照 判定规则计算将每个视觉块的DoC值并与预设PDoC值进行比较，如果 DoC<PDoC,则表示该视觉块无需继续分割，否则对该视觉块递归地执行相同的 分割过程，直到获得足够小的DoC值大于预设PDoC的视觉块或视觉块为基本 对象不可再分。

基于W3C HTML网页设计规范，VIPS算法按优先级定义了 13条启发式判 定规则，在判断当前节点是否需要继续拆分时，对于不同类型的节点算法采用不 同的规则组合［灼。

提取所有视觉块后，第二步是将它们放入池中以进行可视分隔条检测。分隔 符是网页中的横向或纵向的矩形，它们在视觉上交叉。分隔条由一个二维向量表 示：（Ps，Pc），其中Ps是左上角像素坐标，Pe是右下角像素坐标。分隔条的宽度 由这两个值之间的差计算。视觉分隔条的检测过程如图2-8所示。



图**2-8**视觉分隔条的检测过程

分隔条检测完成之后是内容结构重构。将分隔条两侧的视觉块合并，每个新 区块的DoC值更新为组成新区块的视觉块的最大DoC值。将每个新区块的DoC 值与PDOC对比，如果DoC<PDoC,那么继续下一轮的迭代过程，直至所有视 觉块的DoC<PDoCo如果所有区块都满足要求，则停止迭代过程，并获得整个 页面的基于视觉的内容结构。

VIPS算法根据节点内聚度将网页划分为多个视觉块，需要配合其他方式才 能提取网页信息。何正华等人闵使用VIPS算法构造视觉分块树，根据通过归纳 的BBS主题网页的视觉特征，定位用户发言区，然后过滤无关的外链附加块， 通过外观相似度聚类视觉块，最后重组构建结构化的BBS用户发言抽取结果。 吴倩等人匸别提出了视觉块定位算法VBPA（Visual Block Positioning Algorithm）,利 用改进的VIPS算法构建视觉分块树，根据纯文本密度与链接文本密度等特征量 定位主题信息，进而提取主题信息。Michael Cormier等人㈣提出了一种基于纯 视觉信息而不是基于实现细节来分析网页结构的方法，通过分析网页布局将其划 分为不同语义结构的视觉块。

基于视觉特征的网页信息提取摆脱了对于文本的和DOM树结构的过度依赖, 同时利用了网页的视觉信息，使得该方法的适用性和准确性也更好【3。】。由于视觉 块提取过程中用到了根据网页设计规范设计的启发式规则，网页设计规范随着技 术的发展一直在变化，而且不是所有的编程人员都遵守设计规范，需要不断地去 归纳总结判定规则。

**2.3.4**基于机器学习的网页信息提取

基于机器学习的网页信息提取算法是通过分析要提取信息项的特征，利用标 注好的网页数据集训练生成网页节点分类器，然后对网页节点进行分类，最终完 成网页信息的提取。此类算法的大致流程如下：

（1）找一定数量的网页作为训练集，对网页进行人工标注。

1. 设计特征。根据网页特点设计特征。
2. 选择合适的分类器，使用上一步设计的特征进行训练。

安增文等人⑶】根据网页生成DOM树，然后将文本密度、链接密度和与标题 相关度等值输入BP神经网络进行训练，建立抽取规则进而抽取网页正文。

吕容政等人提出一种基于决策树分类的网页正文抽取算法,对文本密集型 网页进行分析，从多个角度对网页正文和噪声进行分析，根据四个特征计算信息 增益，建立决策树模型，将DOM树节点分类为正文与噪声，信噪比最高的DOM 节点即为网页正文区域。这种方法解决了阈值需要通过人工大量实验来确定的问 题。

S. Debnath等人"设计了--种新的算法将HTML页面划分为组成Web页面 块，然后通过查找在网页上不会大量出现的块以及查找具有所需功能的块和使用 经过分类训练的分类器的方法来标识主要内容块。

Shanchan Wu等人図提出一种结合机器学习和分组技术的网页信息自动提 取算法，利用DOM树节点属性生成多个特征以训练机器学习模型，然后根据模 型选择候选节点，开发了分组技术过滤掉候选节点中的噪音并挑选出候选节点缺 失数据。

基于机器学习的方法能处理大规模的网页，但抽取效果取决于数据集的质量 咼低。

2.4分布式系统技术概述

分布式系统与单机系统的不同在于解决的问题的规模，通过协调多台计算机 同时工作，分布式系统可以解决单台计算机难以解决的大型问题，即计算量和数 据量特别大的任务。分布式系统需要拆分数据和任务，协调系统中的每个节点处 理原任务的一个子集，这涉及到分布式数据存储和任务调度，下面对分布式数据 存储和任务调度的相关理论和技术进行概述，同时介绍分布式系统的设计方法。

2.4.1分布式数据存储

分布式系统数据存储主要分为数据在系统中分布方式、数据一致性和数据备 份三部分。

数据在系统中通常有哈希值分布、数据量分布、数据范围分布和一致性哈希 四种分布方式。按哈希值分布的方式是通过将数据哈希取模分布在不同的节点上, 优点是简单易实现，缺点是节点规模扩展时数据需要迁移并重新分布，而且可能 会出现数据分布不均造成的“数据倾斜”问题。按数据范围分布是将数据按特征 值的值域范围进行划分。按数据量分布不存在“数据倾斜”问题，缺点是需要记 录复杂的数据块分布情况。按一致性哈希的数据分布是改进按哈希值分布的方式, 将节点按性能分为多个虚拟节点均匀分布到一致性哈希值域环上。

数据一致性是指在任意时间，任意节点看到的同一个数据是完全一致的。分 布式系统中各节点分工协作，每个节点负责任务的一部分，带来了数据一致性问 题，每个节点“看”到的数据可能是不相同的。数据不一致的可能是由程序错误、 硬件错误或者网络错误产生。考虑时间维度时，不存在真正意义上的数据一致性， 根据对数据一致性的要求不同，可将数据一致性分为弱一致性、最终一致性和强 一致性。

当发生故障或错误时，通过共识算法解决数据不一致的问题。共识算法的本 质就是纠错，常用的共识算法有实用拜占庭容错(Practical Byzantine Fault Tolerance, PBFT)算法、工作量证明(Proof of Work, PoW)算法、Raft(Reliable, Replicated, Redundant, And Fault-Tolerant)算法等。



Consistency

Availability



Partition

Tolerance

图2-9 CAP理论

分布式事务是数据发生变更时，保证所有相关数据达到预期的过程或手段。 CAP(Consistency, Availiablity, Partition-tolerance)理论如图 2-9所示，” Consistency" 指系统内数据强一致性，”Availiablity”指系统在出现异常时服务对外是可用的，” Partition-tolerance”指系统可以对不同分区进行容错处理，节点分布于多个网络 分区中，系统能够解决分区间通信失败造成的问题。

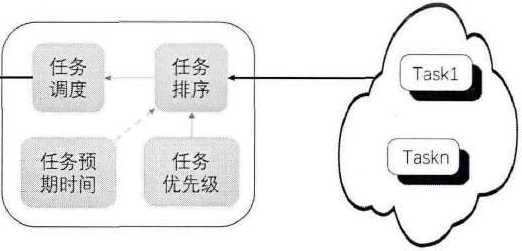
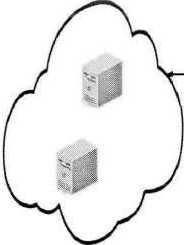
CAP理论指出不存在分布式系统同时具备三个属性，只能在这三者间权衡 取舍。分布式系统中使用分布式事务在短时间内达到数据最终一致性，放弃强一 致性，目前业界主流的方案有两阶段提交、三阶段提交和TCC(Try, Conform, Cancel) o

数据备份是提高分布式系统容错率和可用性的有效手段。将数据按某种方式 拆为较合理的数据段，以数据段为数据备份的基本单位作为副本分布在集群中的 其他节点中。与以机器节点作为副本单位的备份方式相比，数据丢失后的恢复效率更高。

2.4.2分布式任务调度

分布式系统应该有良好的任务调度策略，这对发挥系统的性能和保持系统的 负载均衡具有非常重要的意义〔35】,能够提高系统资源的利用率与任务调度的效率。 任务调度包含两个问题：什么时间调度任务，什么任务应该被哪台机器执行。根 据任务之间的依赖关系可将任务调度分为两类：独立任务调度和相关任务调度。

在分布式环境下，n个独立任务之间没有通信开销和依赖关系，他们需要 竞争使用m个计算资源节点，任何一个独立任务均可以在任何计算资源节点上 运行，独立任务不能被拆分，没有完成的任务不允许被中断。独立任务调度模型 如图2-10所示。独立任务调度算法包括：轮转调度算法、最小链调度算法、最 少执行时间调度算法、粒子群调度算法以及基于一致性哈希的调度算法等。



计算资源集

任务集

图**2-10**独立任务调度模型

王霓虹等人㈣在一致性哈希算法的基础上提出了■-种基于SP-cycle的任务 调度算法，通过构造环形序列取得了更好的负载均衡性能。

相关任务调度是指任务间存在依赖关系，只有当前任务依赖的任务执行完成 后，才能调度当前任务SI。相关任务调度算法有表调度算法、基于任务复制的调 度算法和利用随机搜索技术的调度算法等。

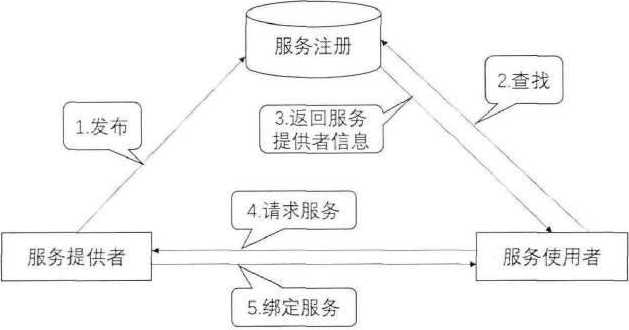
Lindong Liu等人网提出了一种任务调度模型和一种基于I-Apriori算法的 TSFC算法，I-Apriori算法生成的关联规则与任务集中每个任务的最短完成时间 结合在一起。Yuqi Fan等人提出了一种基于任务完成时间预测的动态优先级列 表调度（DPLS）算法，该算法根据先前调度的任务的服务器分配动态预测要调 度的每个任务的剩余执行时间，并基于先前的任务调度结果来确定要调度的下一 个任务和服务器分配。

在现有的分布式系统中，任务调度方式多为集中式调度，即存在一个调度中

心负责整个系统的任务调度，随着任务量的增加，集群需要横向扩容，节点数量 的增加使得主节点的计算能力和通信能力在进行任务调度时成为瓶颈，导致系统 扩容后反而任务处理能力下降。目前出现了一些分布式环境下无中心的任务调度 方法，房婷"提出了基于多Agent的任务调度模型，通过多Agent的合作交互共 同完成任务的协商。

2.4.3面向服务架构

面向服务架构(Service-Oriented Architecture, SOA)是--种分布式运算的软 件设计方法，服务是提供业务功能的独立单元，将系统拆分为多个服务使用一种 统一和通用的方式进行交互。面向服务架构具有松耦合、伸缩性强、易维护等优 点。SOA的工作原理如图2-11所示。



图**2-11 SOA**中不同组件及其工作流程

经过多年的SOA探索，在面向服务架构基础上产生了微服务架构。其基本 思想是围绕业务来有效的拆分和创建应用，实现应用快速开发和部署。

2.5本章小结

本章首先对网络爬虫技术的概况和网页信息提取技术的相关研究进行了概 述，最后介绍了分布式系统中的数据存储、任务调度技术和系统设计方法。本章 的主要内容为后面几章研究网页信息自动提取算法和设计实现分布式爬虫系统 奠定了基础。

第三章文章型网页信息自动提取算法研究

3.1引言

本章研究如何自动提取资讯类、知识类、政策法规类等文章类型网页中的元 数据，如标题、正文、作者、发布时间、图像和附件等信息。从网页DOM树角 度，可以将网页信息提取理解为DOM树节点分类或标注问题，将DOM节点分 类或标注为某项元数据或者噪音。为了简化问题，提高信息抽取的准确率，通常 先合并或者删除部分DOM树节点，然后利用启发式规则或者机器学习的方式对 剩余节点进行分类或标注。

合并节点的过程就是网页分块的过程，指利用视觉信息将多个类型相同的节 点合并为一个节点块，也称视觉块，“类型”指节点对应的元数据类型。VIPS算 法是一种经典的网页分块方法，具有较好的节点合并效果，上文在介绍此算法时 提到其合并过程中使用了 HTML标签，网页设计风格变化导致部分HTML标签 不再使用，使其不再适用于当前网页，而且VIPS算法制定的规则十分复杂，算 法效率低。孙景春⑶等人结合目前流行的网页设计风格对viPS算法进行了改进， 但仍存在着算法效率低和网页设计风格变化导致的节点合并效果差的问题，针对 此问题，本文提出了不依赖于网页设计风格的基于视觉块一致性的网页分块算法, 其适用性更强。

删除节点指利用文章型网页的特征过滤掉网页文章主体所在区域前后的噪 音节点，目前的研究大部分集中于网页正文提取，即只提取文章的正文，忽略了 正文前后的其他信息，本文利用统计特征和视觉特征定位网页主体区域，保留正 文前后的其他有用信息，由此造成的噪音节点过滤不完全问题在下一步解决。

对剩余节点进行分类或者节点标注提取网页信息，部分研究人员使用启发式 规则对节点分类，如区基伟Hi】等人，部分研究人员通过机器学习的方式对节点分 类，如S. Debnath【33】、ShanchanWu^l等人。但这些研究存在着未充分利用上下 文信息和提取精度不足的问题，条件随机场模型可以把上下文信息融入转移特征, 结合文本、视觉和字典等三类20个状态特征，本文使用三阶条件随机场模型进 行序列标注，完成网页元数据的提取，提高了信息提取准确率。

3.2文章型网页分析

本文对文章型网页进行有效信息提取，文章型网页通常包含大量的文字，主 体的周围混杂着大量的噪音信息，如导航栏、广告链接、版权信息等。本文定义 文章型网页包含的有效信息为文章的标题、正文、作者、来源、发布时间、图像 和附件等。标题是反映文章内容与主旨的简短语句；正文是一篇文章的正式文本, 它是文章的主要组成部分⑷)；作者是创作文章的人；来源和发布时间是文章的出 处和发表时间；图像是用于辅助说明文章内容的图片；附件是插入在文章中间或 末尾用于辅助说明的文件。图3-1为文章模型。

—翎

—发布暗间

—作者

**文章——**

—回

—鹿片

—附件

图**3-1**文章模型

文章型网页的特点可以归类为视觉特征、文本特征和DOM树特征，下面将 从以上三个方面进行详细分析。

网页的视觉特征是人们理解页面语义结构最重要的线索，通过这些特征，人 们可以理解网页的组织形式和识别页面传递的有效信息，将网页划分为不同的视 觉块【4我以下是文章型网页的视觉特征：

1. 位置特征。不同的网页元素在页面上的位置不同，内容连贯或相近的 节点通常是连续的且在页面上左对齐。位置特征由节点左上角的坐标来确定。
2. 尺寸特征。尺寸特征由节点的大小来确定，也就是节点的宽度和高度。 同一■内容所在的节点一般具有相同的宽度；单个噪音节点通常只有一行文本，髙 度较小。
3. 格式特征。格式特征为节点中文本的字体属性，包含字体样式、字体 颜色、字体大小和字体是否加粗等。

部分视觉特征是相对的，如节点的尺寸和字体的大小，不同的网页间可能差 别较大，可用网页内部相对大小准确表达其视觉特征含义。

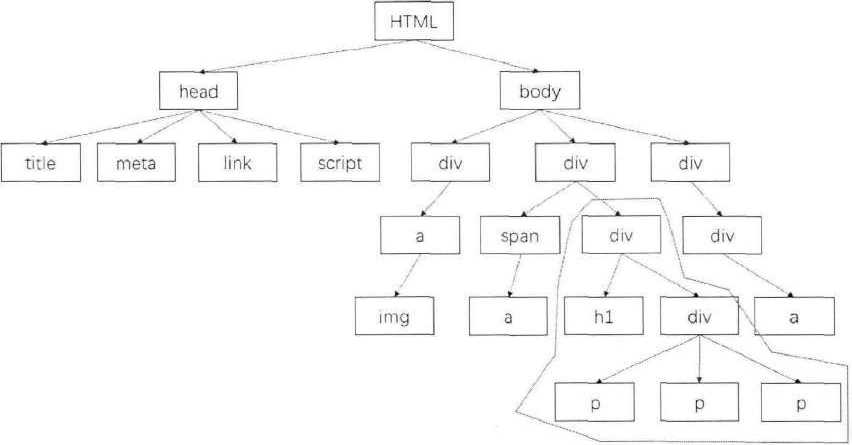
文章型网页的有效信息部分包含大量的文字，噪音部分包含大量的超链接， 基于此，提出如下文本特征：

1. 文本密度。正文文本包含大量的文字且集中在网页的某个区域，文本 密度定义为网页文本与节点数量的比值，不同网页区域文本密度差别较大。
2. 符号密度。在某一文本块中，将标点符号数量与文本块内字符总长度 之比称为该文本块的符号密度(43】。正文文本包含大量的文字的同时也包含了较多 的标点符号，尤其是句号，噪音文本和标题等信息通常不包含或包含少量符号， 一般不包含句号【的。
3. 超链接密度。一般来说正文文本很少包含超链接，而噪音区域包含了 大量的超链接，而且很大比例都是超链接文本。

网页结构可以解析为DOM树，如图3-2所示，文章型网页在DOM树结构 上有如下特征：

(1)文章文本对应的节点在DOM树上通常是相邻连续的，如图中所示。

(2)大部分网页head信息中的网页标题title和文章标题是一致的，meta 信息中作者和文章作者是一致的。



图**3-2**简化的文本**DOM**树

3.3基于视觉块一致性和序列标注的文章型网页元数据提取算法

网页主体内容提取是从网页中选择•一组信息区域，网页元数据提取是从网页 中选择一组信息区域并将不同区域分类标注。网页由HTML语言编码，可以解 析为DOM树结构，图3-2展示了一个简化的DOM树，网页元数据提取可以表

述为DOM树节点选择与标注问题。网页主体内容提取实质上就是解决节点选择 问题，但目前的网页主体内容提取研究只考虑文章型网页正文，忽略了标题、作 者、发布时间、来源等其他元数据，同时准确率有待进一步提高，本小节提出的 网页分块与主体区域定位的网页主体内容提取算法解决了以上问题。在节点选择 完成之后使用CRF实现节点分类标注，完成网页元数据的提取。网页信息提取 算法流程图如图3-3所示。

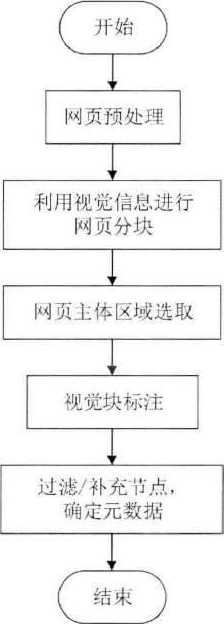


图3-3网页信息提取算法流程图

3.3.1网页预处理

网页预处理分为两部分：标准化网页格式和清除冗余信息〔4习。

HTML语言规范对于语法部分的要求相对较为宽松，这导致了部分网页在源 代码上的不规范性W】。这些不规范主要表现在标签对缺失、标签嵌套错误和特殊 字符未转义，这些错误会对网页信息提取的正确性和效率造成阻碍。通过对网页 格式标准化可以解决这个问题。

清除冗余信息是指删除网页源码中不必要的标签及内容，包括注释、脚本、 样式和互动标签，本文使用正则表达式匹配的方式清除冗余信息，表3-1展示了 使用的正则表达式。

表3-1网页预处理正则表达式

|  |  |
| --- | --- |
| 正则表达式 | 含义 |
| (?is)<!-[A>]\*-> | 匹配HTML注释 |
| (?is)<script. \* ?>. \* ?</script> | 匹配＜script＞标签 |
| (?is)<style.\*?>.\*?</style> | 匹配＜style＞标签 |
| (?is)<\s\*link[A>]\*> | 匹配＜link＞标签 |
| (?is)<input.\*?> | 匹配＜input＞标签 |
| (?is)<select.\*?>.\*?</select> | 匹配〈select〉标签 |

3.3.2网页分块与网页主体区域定位

典型的页面分块算法VIPS及其改进算法被广泛应用，然而VIPS算法存在 两个问题：一是过度依赖HTML标签，VIPS算法在提取视觉块时定义了 13条 判定规则，这些规则针对不同的HTML标签有不同的处理方式，网页设计风格 在不断变化，当前主流网页设计釆用的WEB设计标准DIV+CSS布局代替了传 统通过表格(table)布局定位的方式，使得某些判定规则不再适用。二是需要预 设PDoC值，对于不同的页面类型和应用要求，这个值往往不同，难以确定ML

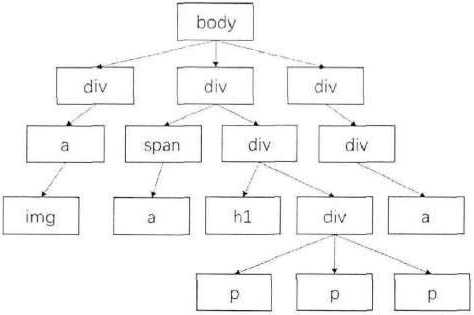
本文提出一种新的页面分块算法，不针对特定的HTML标签，根据节点的 视觉特征，结合DOM树结构将网页划分为多个一致性视觉块，为后续的目标数 据区域定位划分出合适的基本处理单元。

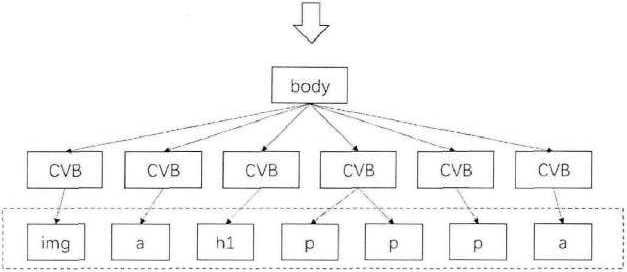
首先介绍一致性视觉块(CVB)的概念，一致性视觉块中的节点应满足以下 条件：

(!) 一致性视觉块如果包含多个节点，节点在页面布局和DOM树中都是 相邻的；

1. 一致性视觉块中的节点是左对齐或者顶端对齐的：
2. 如果一致性视觉块中的节点左对齐，那么节点的宽度需要保持一致； 如果一致性视觉块中的节点顶端对齐，那么节点的高度需要保持一致：
3. 一致性视觉块中各节点的字体是相同的，包含文字字体、文字大小、 文字颜色、是否加粗和是否有下划线；
4. 不含文本的每个节点都是单独的一致性视觉块。

基于节点特征的页面分块算法将输入的网页DOM树转化为页面的所有一致 性视觉块输出，如图3-4所示。输出只有根节点和叶子节点，每个叶子节点都是 一致性视觉块，虚线框仅为示意，表示一致性视觉块包含的节点。





图**3-4**页面分块算法的输入输出

基于视觉块一致性的页面分块算法伪代码如表3-2所示，主要流程如下：

1. 遍历DOM树，获取叶子节点的列表List；
2. 对List中每一个叶子节点，判断其是否为文本节点或图像节点，不是 则删除该叶子节点；
3. 査找List中叶子节点的兄弟叶子节点，判断相邻兄弟节点与该节点是 否符合一致性视觉块的标准，如果不符合，将该叶子节点判定为独立的一致性视 觉块，如果符合，将两个叶子节点合并为一个一致性视觉块，继续判断相邻兄弟 节点与该一致性视觉块的一致性，以此类推，将全部兄弟叶子节点判定完毕；
4. 一个叶子节点与其兄弟节点合并完毕后，如果有父节点，将合并后的 一致性视觉块替换父节点，每个视觉块都将作为一个新的叶子节点，替换后一致 性视觉块中的节点仍要保持其原来的顺序；
5. 重复执行上述步骤，直到所有叶子节点均为一致性视觉块节点。 在完成网页分块后，下一步是对网页主体区域定位，在所有的一致性视觉块

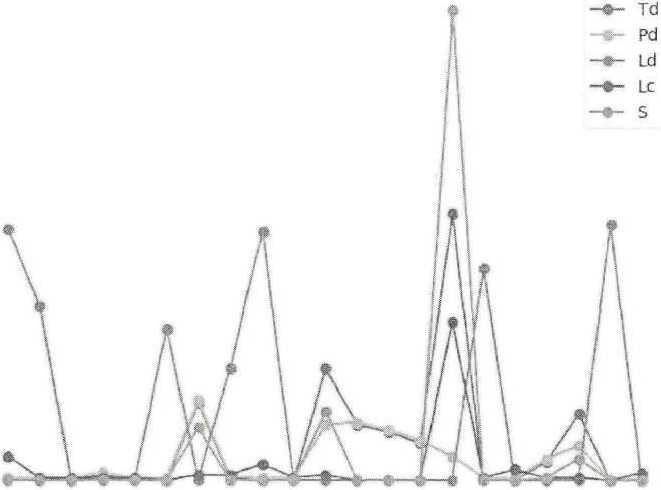
中选出网页主体视觉块，同时也是在网页DOM树中选出网页主体对应的节点。

本文针对文章型网页进行信息提取，文章型网页主体的特征前文己经分析过, 本文将根据文本密度、符号密度、超链接密度和视觉块文本行数将一致性视觉块 分为内容视觉块和噪音视觉块。

表3-2页面分块算法

|  |  |
| --- | --- |
| 算法：基于视觉块一致性的页面分块算法 | |
| 输入： | 网页DOM树 |
| 输出: | 一致性视觉块树CVBT |
| *C* reateCVBTFromDOM(DOM\_tree): | |
| 1 While DOM\_tree.depth > *2:* | |
| 2 | 遍历获取所有的叶子节点，组成叶子节点列表nodeQueue |
| 3 | Foreach node in nodeQueue： |
| 4 | If node is not文本节点or node is not图像节点： |
| 5 | 将node从DOM树删除 |
| 6 | 将 node 从 nodeQueue 删除 |
| 7 | break |
| 8 | End If |
| 9 | 将 node 放入 tempCVB |
| 10 | 获取node的兄弟叶子节点SiblingQueue |
| 11 | While SiblingQueue.size > 0: |
| 12 | If相邻的兄弟节点nextSiblingNode与node符合一致性视觉块 |
|  | 标准： |
| 13 | 将 nextSiblingNode 放入 tempCVB |
| 14 | 下一个相邻兄弟节点为nextSiblingNode |
| 15 | 将 nextSiblingNode 从 SiblingQueue 删除 |
| 16 | Else： |
| 17 | 将 tempCVB 加入 tempCVBList |
| 18 | 下一个相邻兄弟节点为node |
| 19 | 将 node 从 SiblingQueue 删除，且放入 tempCVB |
| 20 | End If |
| 21 | End While |
| 22 | 将 tempCVB 加入 tempCVBList |
| 23 | parent = node的父节点 |
| 24 | If parent is not NULL AND parent is not body： |
| 25 | 将tempCVBList中~致性视觉块作为叶子节点替换parent |
| 26 | End If |
| 27 | End Foreach |
| 28 End While | |

下面将介绍文本密度、符号密度、超链接密度和视觉块文本行数的计算方式。 图3-5为某文章型网页各特征数值折线图。



175 -

150 -

125 -

100 -

75 -

50 -

25 -

0-

0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5 20.0

图**3.5**某文章型网页各特征数值折核图

文本密度。如果一个视觉块包含n个节点，那么该视觉块的文本密度以为 支(”-化)

*Td=* (3-1)

其中：叫为节点i的文本字数，为中文字符数量与英文单词数量之和 *It,*为节点i中超链接的文本字数。

符号密度。将字符总长度与标点符号数量之比定义为符号密度，符号密度Pd 的计算公式为

Pd =

**1+**万

1=1

其中，日为节点i中的符号数量。在一段文字中，正文文字越多，标点符 号也会越多，文字的增长显然比标点符号的增长快得多，所以文章正文部分符号 密度Fd值相比于噪音部分会比较大。

超链接密度。将超链接文本长度与字符总长度之比定义为超链接密度，超链 接密度的计算公式为

n

*Ld=* — (3-3)

1+£叫

/\*!

其中4为每个节点中超链接文本的长度。

视觉块文本行数。视觉块文本行的数量与显示器分辨率有关，网页设计时需 要考虑对不同分辨率浏览器的兼容性，目前网页设计的主流宽度都在960px以上, 本文在获取网页的视觉信息时将浏览器宽度设置为960px。视觉块文本行数的计 算公式为

% \_ & height - padding \_ top \_ padding \_ bottom (34)

C line \_ height

其中，分指节点高度，

pai灿成 伽指节点内元素内容与元素边框的上边距， *padding bottom*指节点内元素内容与元素边框的下边距， *line height*指多行元素的空间量，即多行文本间的间距。

基于以上特征，对视觉块进行选择：

1. 遍历所有视觉块，计算每个视觉块的文本密度以、符号密度用、超 链接密度M和文本行数左；
2. 在所有未标记视觉块中找出文本密度"最大的视觉块，判断其文本 行数聚是否大于3,若是将该视觉块作为基准视觉块执行下一步，否则 将该视觉块标记重新执行此步，如果连续3次判断失败就认为该网页不 是文章型网页；
3. 从基准视觉块向上查找网页主体部分的开始视觉块，通常文章标题的 行间距是最大的，向上找出行间距最大且文本行数化不大于2的视觉块, 从该视觉块开始使用滑动窗口的方式向上计算3个相邻的视觉块的平均 超链接密度，如果平均超链接密度大于20%则停止，将停止时的视觉块 作为开始视觉块，下一步执行步骤；如果找不到满足行间距最大且文本 行数M不大于2的视觉块，执行下一步；
4. 从基准视觉块开始使用滑动窗口的方式向上计算3个相邻的视觉块 的视觉块得分均值S和平均符号密度，S的计算方式为式3-5,直到计 算结果S小于10时停止，此时使用与步骤＜3)中相同的平均超链接密 度判断得出开始视觉块；

*S = Td\** log(HNodeNum)\* log(Pd) (3-5)

式中NodeNum为节点数量

1. 结束视觉块的判定方式同步骤(4)中开始视觉块的判断方法。
2. 开始视觉块与结束视觉块之间的视觉块判定为网页主体区域。

本小节利用视觉信息，自下而上的将网页分为多个一致性视觉块，结合文本 统计特征和部分视觉特征将一致性视觉块划分为连续的主体块和噪音块，从原始 DOM中选择出网页主体对应的节点。通过这一步，缩小了网页元数据的提取范 围，提高了网页元数据提取的质量。

3.3.3基于CRF的网页元数据抽取

在网页数据抽取过程中，对网页DOM树节点进行预测标注时，当前节点的 状态不仅与其本身的特征有关，还依赖于节点在DOM中的位置，即节点的上下 文环境。CRF可以对这种复杂的相互依赖关系进行建模，在进行信息识别与抽 取时，具有更高的准确率和召回率，近年来已经成为信息抽取领域用得较多的一 种模型

条件随机场是给定一组输入序列条件下预测另一组输出序列状态的条件概 率分布模型【48)：

设X与：T是随机变量，*P(Y\X｝*是给定X时Y的条件概率分布，若随机变 量V构成的是一个马尔科夫随机场，即满足

P(K |X占」,土匕，…,*Yn) = P(Yl\X,Yi\_l,Yl+i)* (3-6)

则称条件概率分布*P(Y\X* )是条件随机场。当X与\*有相同的结构时就构 成了线性链条件随机场，具有如下形式：

*P(y* I x) = exp(Z &匕(M，乂，x，0 *+£ 卩向(乃,x,i)) (3-7)*

其中，

Z(x) = £ exp§ 从+Z *收*(3-8)

*y '.k 'J*

*tk*是定义在边上的特征函数,称为转移特征，依赖于当前位置和前一个位置, 号是定义在结点上的特征函数，称为状态特征，依赖于当前位置。条件随机场由 特征函数孔和号与其对应的內和巧确定。

本文使用CRFI具CRFsuit训练CRF模型，相比于其他工具具有训练速度 快和训练测试文件的格式简单等优点，实现了 L-BFGS、OWL-QN和AROW等 优化算法，提升了工具进行模型训练的效率以及预测性能。

CRFsuit的主要功能是训练条件随机场模型以及使用模型进行标注，训练的结果为一个模型文件，使用模型文件可以对数据进行标注。在训练之前需要确定 状态标签和特征。

与文章型网页元数据相对应，模型抽取内容所对应的状态标签及说明如表 3-3所示，其中多了噪声信息标签<noise>,因为上一步选中的网页主体很可能会 包含网页噪音。

|  |  |
| --- | --- |
| 表3-3状态标签及说明 | |
| 状态标签 | 描述 |
| <title> | 文章标题 |
| <author> | 文章作者 |
| <publicTime> | 发布时间 |
| <publisher> | 文章来源 |
| <mainText> | 文章正文 |
| <picture> | 图片 |
| <attachment> | 附件 |
| <noise> | 噪声信息 |

特征的选择是机器学习的关键，合适的特征对提取性能有较大提升〔49】。本文 选取三类共20个特征，分别是文本特征、视觉特征和词典特征。

文本特征是指一致性视觉块中文本文字的特征，所用的文本特征如表3-4所

zj\o

表3-4史本特征

|  |  |
| --- | --- |
| 特征 | 含义 |
| AllDigitsOrChar | 是否均为数字或标点字符 |
| ContainsPeriod | 是否包含句号 |
| TextDensity | 文本密度 |
| CharDensity | 标点符号密度 |
| LinkDensity | 超链接密度 |

视觉特征是指一致性视觉块的格式和布局，本文所用的视觉特征如表3-5所

Zjx 0

表3-5视觉特征

|  |  |
| --- | --- |
| 特征 | 含义 |
| BlockFontSize | 字体大小 |
| BlockFontColor | 字体颜色 |
| BlockFontBold | 字体是否加粗 |
| BlockFontltalic | 字体是否倾斜 |
| BlockLineCont | 文本行数 |
| BlockWidth | 视觉块宽度 |
| BlockCenter | 视觉块中心 |

表3-6词典特征

|  |  |
| --- | --- |
| 特征 | 含义 |
| ContainsArticle | 属性是否含有article |
| ContainsTitle | 属性是否含有title |
| ContainsAuthor | 属性是否含有author |
| ContainsTime | 属性是否含有time、date |
| ContainsSource | 属性是否含有source |
| ContainsPublic | 属性是否含有public> publish |
| ContainsContent | 属性是否含有content |
| ContainsAttachment | 属性是否含有 txt、pdf、word、excel、 rar、zip |

词典特征是指节点属性中是否包含特殊标记，如title、author, time等关键 词，节点的特殊标记通常与节点的标签有重大关联。本文所用的词典特征如表 3-6所示。

在确定标签和特征后生成训练文件，抽取特征时同时抽取当前视觉块位置及 前两个视觉块的特征，当前位置的特征为状态特征，与前两个位置特征的组合为 转移特征，特征权值的确定过程就是三阶条件随机场模型的建立过程。特征抽取 伪代码如表3-7所示。本文在训练时将L-BFGS训练算法与Elastic Net (L1 + L2) 正则化配合使用求解三阶条件随机场模型的参数，L-BFGS算法收敛性较好、计 算速度快，Elastic Net (L1+L2)算法在损失函数后添加L1正则项和L2正则项 可以解决模型训练过程中的过拟合问题。在预测标注过程中利用Viterbi解码来 得到最优序列。

表**3-7**特征抽取伪代码

**算法：特征抽取**

输入：视觉块列表blocks, block的位置顺序i

输出：特征字典features

**block2 features(blocks, i):**

1. block = block[i]
2. features = {" feature 1": block, feature 1,...}
3. Ifi>0:
4. block 1 = block[i -1]
5. features.update(("-l rfeaturel":block 1 .feature 1
6. End If
7. Ifi> 1:
8. block2 = block [i - 2]
9. features.update(("-2:featurel":block2.featurel,...})
10. End If
11. Return features

3.4实验分析

为了验证本文提岀算法的提取效果，通过实验对其进行验证并将其与现有文 章型网页信息提取效果较好的算法进行比较。

3.4.1**数据集及评价标准**

本文选取抓取的新浪新闻、中国教育部、英国教育部等国内外网站的400 个文章型网页人工标注后作为数据集进行实验，将其中320个网页作为训练集， 80个网页作为测试集A,另外标注40篇腾讯新闻和CNN新闻网页用作测试集 Bo

本文釆用准确率P、召回率R和综合指标F1对网页元数据的抽取效果进行 评估。

准确率是评价提取整体准确度的指标，标签S准确率*Ps*的计算公式为：

P,=

TP,

TP' + FPs

(3-9)

其中，If表示提取方法在测试样本上标签S标记正确的数量,

*FPX*表示视觉块实际不是标签S,提取方法标记为标签*S*的数量。

召回率是正确标注的标签占该类标签总数的比例，标签S召回率的计算 公式为：

(3-10)

其中，/W,表示视觉块实际为标签S,提取方法标记为非S的数量。

F1综合指标综合评价准确率和召回率，综合评价提取算法提取性能。计算 公式为：

(3-11)

3.4.2实验结果与分析

在测试集上采用本文提出的算法进行信息提取，实验结果如表3-8和表3-9

所示。

表3-8测试集*A*信息提取结果

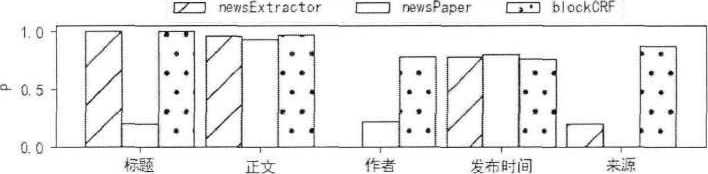
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 标签 | 总数量 | 识别数量 | 正确数量 | P | R | F1 |
| 标题 | 80 | 80 | 80 | 1.000 | 1.000 | 1.000 |
| 正文 | 473 | 494 | 466 | 0.943 | 0.985 | 0.963 |
| 作者 | 50 | 40 | 40 | 0.800 | 1.000 | 0.889 |
| 发布时间 | 80 | 91 | 62 | 0.775 | 0.681 | 0.720 |
| 来源 | 50 | 52 | 46 | 0.920 | 0.885 | 0.902 |
| 图片 | 12 | 14 | 12 | 1.000 | 0.857 | 0.923 |
| 附件 | 22 | 21 | 21 | 0.955 | 1.000 | 0.978 |

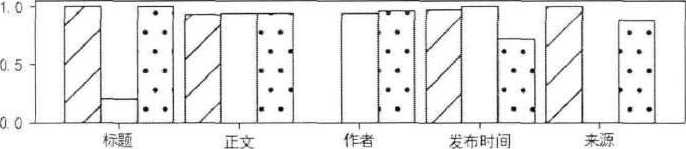
表3-9测试集B信息提取结果

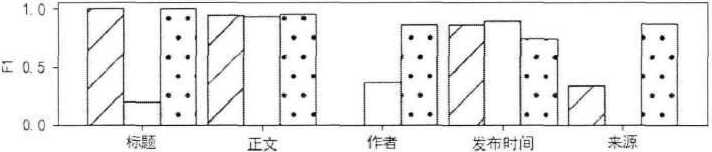
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 标签 | 总数量 | 识别数量 | 正确数量 | P | R | F1 |
| 标题 | 40 | 40 | 40 | 1.000 | 1.000 | 1.000 |
| 正文 | 264 | 259 | 246 | 0.932 | 0.953 | 0.943 |
| 作者 | 40 | 34 | 30 | 0.750 | 0.882 | 0.815 |
| 发布时间 | 40 | 28 | 28 | ().750 | 1.000 | 0.720 |
| 来源 | 40 | 34 | 30 | 0.750 | 0.882 | 0.815 |
| 图片 | 104 | 126 | 104 | 1.000 | 0.825 | 0.905 |
| 附件 | 0 | 0 | 0 | - | - | - |

在测试集A的实验中，本文提出的抽取方法对这些文章页面的标题、正文、

图片和附件都具有良好的效果，这是因为这四项特征明显，对作者抽取的准确率 和召回率都略低，通过对实际页面的分析发现部分作者信息与文章正文放在一起, 与正文的样式并无区别，导致部分识别错误，对发布时间抽取的准确率低，分析 发现部分网页包含发布时间和更新时间，算法对样式相同且位置相近的二者无法 进行有效区分，导致准确率较低。测试集B与测试集A不是同源的网页集合， 通过对比表3-8与表3-9可以发现，使用训练集训练出的模型放在其他网页集上 也有较好的抽取结果，本文提出的信息提取算法有一定的通用性。







图**3-6**文章型网页信息提取方法对比

图3-6是本文提出的算法与现有提取效果较好的算法进行对比。 newsExtractor是文献［51］中的提取方法，提取的内容包括标题、正文、发布时间 和来源。newsPaper是文献［52］中使用的提取方法，提取的内容包括标题、作者、 发布时间、关键字和正文。blockCRF为本文提出的方法，提取的内容包括标题、 作者、发布时间、正文、来源、图片和附件等。本文对3种算法提取的公共数据 项作对比，从对比数据可以看出，newsExtractor和blockCRF提取文章标题的F1 值达到了 100%, newsPaper由于直接提取网页＜title＞标签中的内容作为标题导致 准确率较低，三种方法对正文的提取效果不相上下，newsExtractor不具有提取作 者信息的功能，newsPaper只能够提取英文网页中的作者，对中文网页的提取效 果较差，本文提出的算法对作者的提取效果最好，newsExtractor和newsPaper对 发布时间的提取效果要更好,newsExtractor对文章来源的提取釆用简单的正则匹 配的方式导致提取效果较差，本文提出的算法对大部分网页的文章来源信息提取 较为准确。综合各项指标来看，本文提出的算法提取效果最好。

3.5本章小结

本章介绍了文章型网页的数据模型，在对文章型网页特征进行详细分析的基 础上提出了一致性视觉块的概念，通过视觉特征将网页分割为一致性视觉块，结 合文本特征实现对网页主体区域的定位，然后使用CRF对视觉块进行标注，提 取网页元数据。本章最后还对提出的提取方法进行了实验，并与同类方法进行了 对比。

第四章分布式爬虫系统的设计与实现

4.1引言

针对现有分布式爬虫存在的问题，本章设计实现了基于网页信息自动提取的 分布式爬虫系统。从功能上讲，无论是单机式爬虫还是分布式爬虫，都可以分为 网页下载、网页信息提取和数据存储三个部分，网页下载和网页信息提取的性能 直接决定了爬虫系统信息获取效率。

分布式爬虫系统能够协调若干个节点同时下载网页，网页下载的速度取决于 每个节点的下载速度、节点的数量和任务调度能力。提高单个节点下载速度的研 究和技术已经十分成熟，很难再有大的提升。对于系统能够容纳的节点数量和任 务调度，本文提出了一种无中心的基于动态反馈的任务调度策略，解决了目前众 多分布式爬虫系统使用集中式调度带来的可靠性低和系统扩容瓶颈问题。

人工编写解析脚本进行网页信息提取在对单一网站的抓取中具有较好的效 果，但在面对多个网站时存在人工编写解析脚本工作量大、效率低的问题，对此 本文引入了文章型网页自动解析算法，能够自动解析文章页面进行信息提取。与 使用解析脚本相比，使用解析算法增加了信息提取耗时，但大大减小了面对海量 网站时编写维护解析脚本的复杂性。

4.2系统需求分析

本系统为企业某信息平台的一个子系统，该信息平台用于互联网文章的釆集、 分析与展示，在设计实现本系统前首先对该信息平台的总体结构做一个阐述，通 过了解该信息平台总体层次结构设计，能够从整体上认识分布式爬虫系统在整个 项目中的重作用，从而更好地分析企业实际需求。

信息平台的架构图如图4-1所示，分为数据釆集层、数据存储层、数据处理 层、数据服务层、数据应用层和平台管理。数据釆集层是平台的基础，使用分布 式爬虫从互联网获取各种类型的数据，数据存储层负责存储平台数据，数据处理 层完成资讯标注、数据清洗、离线分析等工作，数据服务层具有全文检索、资讯 推荐、任务调度等功能，数据应用层负责资讯的展示等，平台管理包含系统监控 告警、用户管理、爬虫项目管理、菜单管理等。



數据应用层

数据服务展f/OZTS 繼为MS 晩二二-q 恣二…'後

Hiznm msssM ggs^as

平台朦涯 孚台女全

笙條緒恭 用户淑 逐安装自管理 沒流审浪宅莪溥

数据处理层

数据存储层

数据采集层

榮平後查

图**4.1**信息平台架构图

本文主要涉及数据釆集的相关工作，从互联网获取大量的资讯类、知识类、 政策法规类文章，是信息平台数据的来源，用于后续的处理分析。结合企业的实 际需要，下面将从功能性需求和非功能性需求两个方面进行分析。

4.2.1功能需求

通过结合具体业务场景，本文设计的分布式网络爬虫系统旨在用户可以通过 简单的爬虫任务配置即可抓取海量分布在不同网站上的资讯类、知识类、政策法 规类等文章页面，整个抓取过程无需用户参与。基于此，对分布式网络爬虫系统 进行了详细的分析，明确提出了以下功能需求：

1. 分布式网页下载。各节点相互协作以较快的速度从互联网网站上下载 网页，网页下载是后续处理的基础。
2. 网页信息提取。网页中含有大量的噪音信息，如导航栏和广告栏等， 爬虫需要能够自动提取文章型网页中的信息，无需人工编写解析脚本。
3. 网页去重。互联网中存在着大量的重复网页，需要对URL和网页内容 进行去重，去除重复、冗余数据。
4. 数据存储。对下载的网页和提取的信息做持久化存储，保存到数据库； 同时存储网页抓取过程中的临时数据。
5. 系统节点管理。为应对不同的抓取任务，系统需要具有横向扩展的能 力，即具有节点动态加入与退出功能，所以系统需具备节点管理功能。
6. 爬虫任务配置。用户只需要简单的操作即可配置、管理爬虫任务。

除上述功能外，网站的爬全率需要达到97%。爬全率是指实际抓取的网页数 量与目标网页数量的比。为达到爬全率的要求需要对抓取失败的网页釆取相应的 措施。

4.2.2非功能性需求

作为功能性需求的补充，除了对系统核心功能的实现之外，在设计过程中还

需考虑一下几个方面:

易用性。提供给用户的操作及配置应尽可能简单。传统的网络爬虫系统，需 要对每个网站编写维护网页信息提取脚本，这要求用户有一定的编程基础。本文 设计的网络爬虫系统基于网页信息自动提取算法，具有自动解析网页的能力，无 需编写维护网页信息提取脚本，没有编程基础的用户也可使用。用户在使用爬虫 系统时只需配置爬虫任务，使用简单易操作。

健壮性，也可以称作可靠性，是指分布式网络爬虫系统的容错能力。一个健 壮性强的分布式网络爬虫系统在遇到异常状况时也能够安全、稳定的运行，持续 不断的抓取网页数据㈣。系统异常包含输入异常导致的系统异常和服务器节点异 常。输入异常是指错误的爬虫任务配置、失效的URL链接等，要求系统能够规 范系统输入并做检査。服务器异常通常是指服务器宕机或服务器其他程序异常导 致的网络爬虫系统异常，当出现服务器异常时需要快速做出响应，及时恢复系统， 同时通过邮件和短信的方式告警反馈。

礼貌性。分布式网络爬虫系统需要严格遵守网站的robots协议佝。其次，需 要考虑抓取频率，抓取频率太高会给网站带来过大的负载，可能会引起网站的瘫 痪、网络阻塞等问题，因此在设计时使用合适的速度访问站点下载网页，在满足 爬取需求时，抓取速度应该尽可能的慢。

负载均衡。分布式爬虫系统是通过多个爬虫节点协作完成目标网站网页的抓 取，实际使用过程中个爬虫节点的机器性能各不相同，集群中个爬虫节点的抓取 能力也就不同，使用合适的任务调度算法给每个节点分配合适的任务数量，让每 个节点发挥最大的性能才能最大化爬虫的抓取效率。

通用性。本文设计的分布式网络爬虫系统不是为了抓取特定目标网站，而是 针对资讯类、知识类、政策法规类等文章页面，需要兼容多个网站，同时分布式 网络爬虫作为网页信息资源获取的强有力釆集工具，它应该可以为各个领域的应 用提供强大的数据来源保障，从而最终为各目标应用服务。因此本文设计的网络 爬虫系统需要具有高度抽象的架构设计，无需过多修改即可用于不同领域的应用 场景。

4.3系统总体设计

本节对分布式网络爬虫系统的整体结构进行设计，如图4-2所示，至底向上 的将系统依次分为数据釆集层、数据解析层、数据存储层、节点接入层和系统管 理层。本文在数据解析层引入了文章型网页自动解析算法，能够自动解析文章页 面，解决了人工编写解析脚本效率低下的问题；在数据釆集层提出了一种无中心

的基于动态反馈的任务调度策略，解决了目前众多分布式爬虫系统使用集中式调 度带来的可靠性低和系统扩容瓶颈问题。



图**4-2**分布式爬虫系统层次结构图

<1）数据釆集层，该层的主要功能是从URL队列中获取URL并进行网页 下载。该层包含了任务调度和网页下载。本文设计的分布式爬虫系统各爬虫节点 间是对等的，每个节点都需要具有独立的决策能力，通过节点间的合作和交互完 成抓取任务的协商调度，对此本文提出了一种无中心的基于动态反馈的任务调度 策略。网页下载是利用HTTP协议访问网站并下载保存网页数据，为了防止网络 爬虫被封禁，下载网页时使用随机IP代理器。数据采集层包含网页下载模块和 任务调度模块。

（2） 数据解析层，该层的主要功能是对下载的网页进行自动解析，从网页 中获取结构化的信息。该层将上文研究的网页信息自动提取算法集成到网络爬虫 中，代替传统的人工编写解析脚本的方式，将网页信息自动提取算法封装为一个 服务，使用 restful 风格的应用程序接口 （Application Programming Interface, API） 对外提供服务，爬虫节点中的网页信息提取部分调用该接口即可完成网页信息的 提取。数据解析层主要包含网页信息提取模块。

（3） 数据存储层，该层的主要功能是对爬取过程中涉及数据的临时性和永 久性存储，包括URL队列、下载的网页集与解析数据、节点状态数据、任务状 态数据和下载的网页指纹数据等。对不同的数据使用相应的数据结构和存储介质。 同时在数据存储层实现网页的去重，分为URL去重和网页内容去重。数据存储 层包含存储模块和网页去重模块。

<4）节点接入层，该层的主要功能是对节点的动态接入和退出进行管理， 实现系统的横向扩展。

（5）系统管理层，该层的主要功能是对外提供服务，包括节点管理，爬虫 任务管理，系统监控告警。节点管理是监控节点的状态，可以根据实际需要将在 线的爬虫节点休眠。爬虫任务管理包括爬虫任务的配置、启动和停止。系统监控告警是指对系统各部分的状态进行监测，发现异常状况及时通过邮件和短信发送 告警信息。系统管理层包含节点管理模块和爬虫管理模块。

本文设计的分布式网络爬虫系统采用混合式的网络拓扑结构，中心节点不是 物理意义上的某个节点，而是指一个中心存储集群，用于存储各个节点的状态、 爬虫任务的状态和URL队列，各爬虫节点间是对等的，在中心存储集群中沟通 各自的状态协同完成抓取任务。

分布式网络爬虫系统的架构图如图4-3所示。系统管理员通过系统管理服务 对爬虫节点集群和爬虫任务进行管理，各个节点协同完成抓取任务。

ad~»inisttatOf

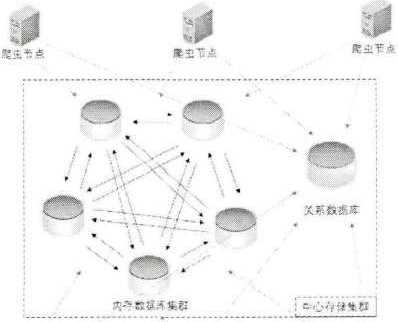
节成2

图4-3

分布式爬虫系统架构图

4.4系统模块设计与实现

4.4.1存储模块的设计与实现



图**4-4**存储结构图

存储模块分为两部分：一部分存储爬虫任务配置以及最终抓取的数据，存储 在关系数据库mysql中；另一部分存储任务执行过程中的抓取信息和节点状态等 信息，存储在内存数据库Redis中，本文使用Redis集群提高性能及可靠性。存 储结构如图4-4所示。两部分共同构成了中心存储集群。

<1）关系数据库表

与爬虫相关的数据库表主要有爬虫任务表crawl\_task、定时任务表 schedule\_task 和爬虫结果数据表 crawl\_webpage。

爬虫任务表crawl\_task存储爬虫任务的配置，数据库表结构如表4-1所示。

表**4-1**爬虫任务表结构

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 描述 |
| id | int | 主键id |
| task id | varchar(32) | 任务id |
| task name | varchar(32) | 任务名称 |
| lask priority | int | 任务优先级 |
| task description | varchar(255) | 任务描述 |
| task\_status | int | 任务状态，0为初始化，1为运行 中，2为暂停，3为停止，4为删 除 |
| is\_schedule | int | 是否为周期任务，0为非周期任 务，1为周期任务 |
| schedule interval | int | 任务间隔，单位为分钟 |
| is\_use\_proxy | int | 是否使用IP代理，0为不使用，1 为使用 |
| is\_download\_interval | int | 是否使用下载间隔，0为不使用，  1为使用 |
| download interval | int | 下载间隔时间，单位毫秒 |
| url\_input\_type | int | 初始url引入方式,1为文件输入，  2为URL配置，3为脚本输入 |
| url detail | varchar(4096) | url引入描述 |
| extern | varchar(4096) | 扩展信息 |
| creator id | int | 任务创建人 |
| create time | timestamp | 任务创建时间 |
| modified time | timestamp | 任务修改时间 |

定时任务表schedule\_task记录定时任务的执行时间及执行状态，详细设计如 表4-2所示。

表4-2定时任务表结构

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 描述 |
| id | int | 主键id |
| task\_id | int | 任务 id,与 crawl\_task 关 联 |
| excute time | date | 执行时间 |
| task\_status | int | 执行状态，0为未执行， 1位执行中，2为执行完 毕，3为执行异常 |
| create time | timestamp | 任务创建时间 |
| modified time | timestamp | 任务修改时间 |

爬虫结果数据表crawl\_webpage用于保存爬虫节点爬取的网页数据，包含 URL链接、网页源码和网页解析数据等信息，详细设计如表4-3所示。

表4-3爬虫结果数据表结构

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 描述 |
| id | int | 主键id |
| page id | varchar(64) | 网页id |
| task id | int | 任务id |
| url | varchar( 1024) | URL链接 |
| url domain | varchar(256) | url域名 |
| webpage source | text | 网页源码 |
| crawle time | date | 网页下载时间 |
| carwler name | varchar(4096) | 网页抓取节点名称 |
| webpage main | text | 网页内容，json格式 |
| parse time | date | 解析时间 |
| parser name | varchar(4096) | 解析器名称 |
| create time | timestamp | 创建时间 |
| modified time | timestamp | 修改时间 |

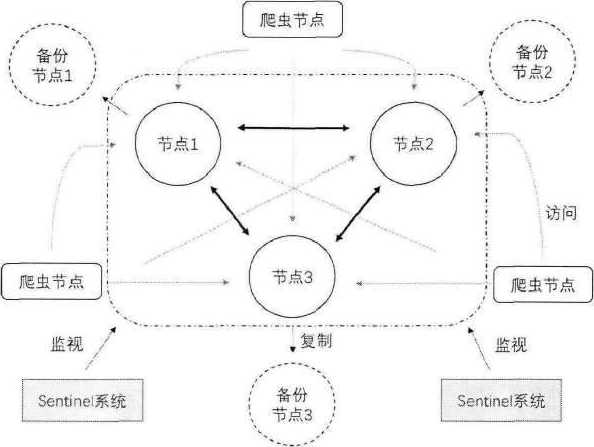
其中，字段webpage\_main是json格式的，包含标题title,作者author,发 布时间public\_time,内容content,图片pictures,附件attachment等信息，对应 内容不存在的字段赋空值。

(2)内存数据库存储

内存数据库存储分为以下单元：URL存储、robots缓存、网页指纹和节点信 息共享。URL存储使用redis中的数据结构sort set实现，分为初始化、待抓取、 定时抓取、临时存储、己完成和抓取失败6个sort set, sort set通过分数对成员 大小进行排序*，*本文将taskjpriority \* 100 000 000 + task\_id作为成员分值， task\_priority为任务优先级,taskjd为任务标识号»Robots缓存使用＜URL\_domain, robots\_content＞形式的键值对存储网站以及对应的robots.txt文件内容，使用redis 中的数据结构string实现，同时设置数据过期时间为1天以应对网站robots协议 的变化。网页指纹存储的设计与实现将在网页去重模块详细说明。节点信息共享 存储单元存储各爬虫节点的信息，包含节点基本信息、节点存活状态、节点服务 器状态和节点任务信息等。

本文中内存数据库集群为redis分片集群，将redis节点连接起来组成一个 分片集群，接收处理爬虫的读写请求，如图4-5所示。分片集群将数据存储在多

个节点，可以极大的降低对单台机器内存资源的占用⑵，同时扩大存储空间。加 入哨兵(Sentinel)系统和备份节点增强redis分片集群的可用性，Sentinel监控



各个节点，当节点不可用时使用其备份节点将其替换,

多个Sentinel避免单个

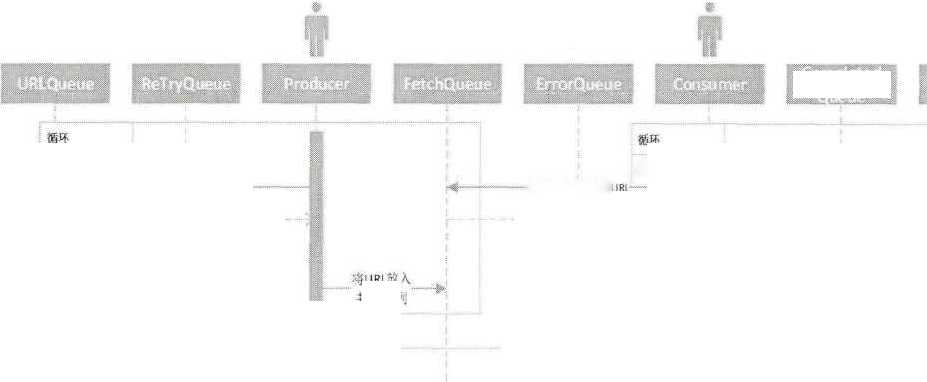
Sentinel故障引发的系统不可靠问题啊。

图4-5redis分片集群

4.4.2**网页下载与任务调度模块的设计与实现**

网页下载主要负责从中心存储集群的待抓取队列中读取URL到本爬虫节点 的下载队列，下载时根据配置决定是否使用随机IP代理，任务调度决定从中心 存储集群读取的待抓取URL的数量。下面叙述如何设计实现网页下载和任务调度。

为增加下载效率，充分利用爬虫节点的性能，网页下载器釆用生产者消费者 模型，有一个生产者和多个消费者，消费者的数量根据爬虫节点所在服务器的 CPU核心数量确定。生产者负责从待爬取URL队列和失败重试URL队列中读 取URL添加至共享队列，失败重试次数超过三次的URL不再放入共享队列，将 被移入失败队列。



根据**URL**域名怜查**robots**缓存

循仄

j下栽**robolsG**议

才 **UR0U** 共享卩、如

**URLQy**技为空裕从失败東讯

'队列•.并取**URL**

**URLQueue** 了为宝

从待爬取臥列读取**URL**

关享认列不

—•从芸亨队刿**15**取**URL**

**Robots**

**Cache**〕

未命中

如果超过重试次数

将如撤入

失职队列

将robols协议按域各放入缓存

超汇最天重试次数放入仙误  
注明下就超时

如果robots协议不允许

爲URL放入

共享臥列

如果F我失敗

下我失般將URL放入堇试"'・壻垂试次散尤1  
将URL及冋页  
放入完成臥列

图**4-6**网页下载模块的时序图

网页下载模块的时序图如图4-6所示。生产者和消费者是不同的线程， URLQueue为爬虫节点本地的待爬取队列；ReTryQueue为失败重试队列，数据 格式为键值对<URL,reTryTimes>,是URL链接和重试次数，最大重试次数为3 次；FetchQueue为共享队列，生产者从URLQueue和ReTyrQueue读取URL链 接后放入该队列，消费者从该队列读取链接进行网页下载操作；EnorQueue是爬 取失败队列，数据格式为键值对<URL,reasonEnum>,是URL和爬取失败原因， 其中 reasonEnum 包括 robotsForbiddenEnum, timeOutEnum, otherEnum, 分别是 robots协议禁止爬取，下载超时和其他原因，ErrorQueue将在每次任务调度开始 之前同步到中心存储集群的爬取失败队列；CompletedQueue是完成下载的队列， 数据格式为键值对<URL,webPageDTO>,是URL链接和网页对象，网页对象 webPageDTO包含网页源码、抓取时间、爬虫节点等信息，后续的网页信息提取 模块将从该队列读取网页对象进行解析；RobotsCache是对网站robots协议的缓 存，用于减少网站robots协议下载次数，提高网页下载效率，RobotsCache缓存 分为两级，一级缓存位于中心存储集群，二级缓存位于每个爬虫节点，设计二级 缓存的目的是减少爬虫节点与中心存储集群的通信次数，新增的网站robots协议 会同步到两级缓存，缓存的数据格式为<URLDomain, robotsContent> , URLDomain指网页域名，robotsContent包含网站robots协议的下载时间和内容， 如果一个网站不存在robots协议，那么robotsContent的值为NULL,为防止robots 协议更新，robots缓存的有效期为1天，缓存过期将会被清理更新。

生产者的工作流程如下：

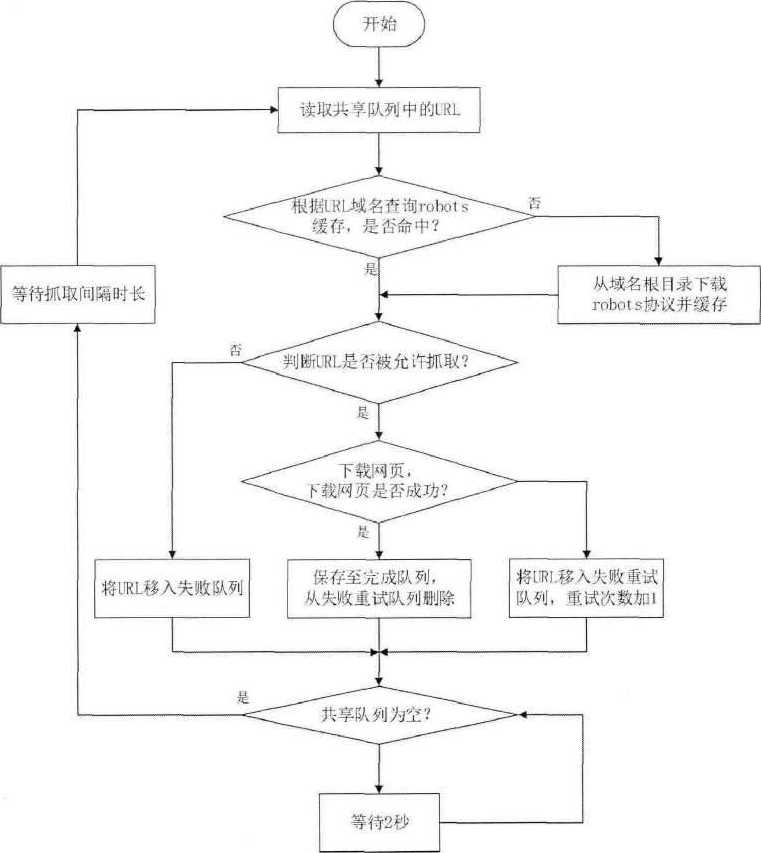
1. 从待抓取URLQueue读取URL链接；
2. 判断共享队列FetchQueue是否已满，若是，等待2秒，重新执行步 骤(2),否则执行步骤(3)；

(3 )将URL放入共享队列FetchQueue ；

1. 判断待抓取URLQueue是否为空，若是，转至下一步骤(5)；否则跳 转至步骤(1);
2. 从失败重试队列ReTryQueue读取URL及其失败次数，若失败次数 不小于3,在失败队列插入键值对<15!<1, reasonEnum〉，reasonEnum为 timeOutEnum,在失败重试队列删除该记录，跳转至步骤(4)；否则将 URL放入共享队列FetchQueue；
3. 判断失败重试队列ReTjyQueue是否为空，若是，等待20秒继续判 断是否为空(防止部分失败重试任务未完成)，若是，转至步骤(7)； 否则跳转至步骤(5)；
4. 生产者休眠，本轮网页下载任务结束。

消费者从共享队列抽取URL链接下载网页，单个消费者的网页下载流程图 如图4-7所示，详细说明如下：

1. 从共享队列读取URL链接；
2. 根据URL域名査询robots协议缓存，如果缓存未命中，去网站根目 录获取并缓存，判断robots协议是否允许URL链接抓取，不允许则将 URL移入失败队列并标注失败原因为不允许抓取。
3. 使用HTTP协议与网站服务器建立连接，建立成功转至下一步骤(4), 否则转(5)；
4. 存储网页数据到抓取完成队列，如果失败重试队列存在该URL则删 除，转(6)；
5. 如果失败重试队列存在该URL则将重试次数加一，否则新增该URL, 且将重试次数设为0；
6. 如果共享队列为空，消费者休眠2秒，再次执行(6)；否则等候抓取 间隔时长，转(1)。



图**4\*7**单个消费者的网页下载流程图

当未配置IP代理且设置抓取间隔时，多线程抓取将改为单线程抓取，即只 有一个消费者。

由于现在部分网站有反爬虫机制，当爬虫节点长时间抓取同一网站的网页时, 可能会出现网站拒绝访问的情况"，所以在设计实现网页下载器时需要考虑这个 问题，本文的解决方法是IP地址代理和随机请求头，使用IP地址代理和随机请 求头，可以让网站服务器认为不是同一个客户端在请求。爬虫使用IP代理可伪 装成不同地理位置的用户，再随机切换HTTP请求头，伪装成不同的用户使用不 同的浏览器访问网站的假象。

即使使用了代理IP地址和随机请求头，网站服务器也可以使用检测IP地址 访问频率和将代理提供商加入反爬虫数据库的方式检测否是一个爬虫，本文通过 设置抓取间隔和自建代理池的方式解决。

随机切换请求头是指在每次HTTP访问时将请求Header中的User-Agent 属性伪装成不同浏览器。主流浏览器的User-Agent信息如表4-4所示。

|  |  |
| --- | --- |
| 表4-4各浏览器的User-Agent信息 | |
| 浏览器 | User-Agent 信息 |
| Chrome | Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36  (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36 |
| Firefbx | Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0)  Gecko/20100101 Firefox/71.0 |
| Microsoft  Edge | Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36 Edg/87.0.664.66 |
| IE | User-Agent:Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0) |
| Opera | User-Agent:Opera/9.80 (Macintosh; Intel Mac OS X 10.6.8; U; en)  Presto/2.8.131 Version/11.11 |
| Safari | User-Agent:Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10\_6\_8; en-us) AppleWebKit/534.50 (KHTML, like Gecko) Version/5.1 Saferi/534.50 |
| 360  Browser | User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; 360SE) |

任务调度负责整个分布式爬虫系统的抓取任务分配。本文提出了一种无中心 的基于动态反馈的任务调度策略，根据爬虫节点与集群的当前状态动态调整任务 分配，实现节点间负载均衡，最大化利用系统资源。本文设计的分布式网络爬虫 系统的节点间是对等的，不存在中心节点，使得集中式的任务调度不再适用，这

就要求每个爬虫节点都要具有独自调度的能力，能够适应系统集群中节点数量的 动态变化。动态调整规则能够实时的评价节点性能和任务调度，从而使得爬虫节 点在任务调度时可以获得更为合适的调度决策。

基于动态反馈的任务调度策略的调度原则为：

1. 任务完成花费时间最小。
2. 负载均衡，要求优先将任务分配到负载较轻的节点。
3. 降低任务调度开销。任务调度开销指任务调度过程中自身节点与其他 节点间信息交互开销和任务调度计算开销，主要表现为任务调度过程中 任务分配后将任务复制到节点的通信耗时和带宽占用，在爬虫系统中具 体表现为分配的URL链接的传输通信量。这要求选择适中的任务分配 量，既不会导致任务分配次数过多造成通信次数过多，又不会导致任务 量过多导致的通信阻塞。

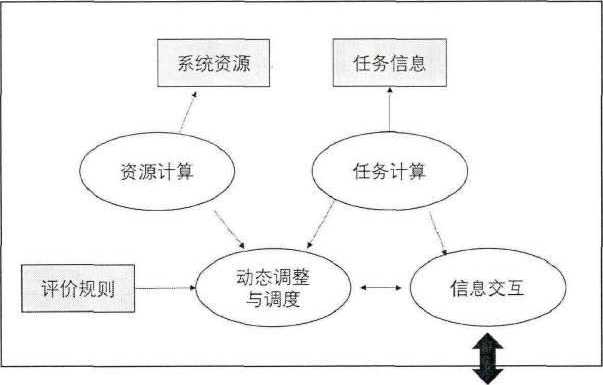


图4-8任务调度模型图

本文设计的基于动态反馈的任务调度模型包括资源计算、任务效率计算、信 息交互和动态调整与调度，任务调度模型结构图如图4-8所示。

资源计算是实时的监测并记录本地爬虫节点的资源信息，如网络带宽、中央 处理器(Centrally Processing Unit, CPU)性能、内存使用状况等，根据资源信息 计算任务执行过程中节点负载值，节点j负载值的计算公式为式4-1。

3

Sj=X]\*m +乂2 \* 〃丿+也 \*</p 且工毛=1 (4-1)

7 /=i

其中，m为内存利用率，与为CPU使用率，方为带宽使用率，带宽使用 率通过一段时间内的网络流量每秒的均值和节点带宽的比值计算得出，耳为资 源系数，不同资源对资源值的影响不同。

任务效率计算是指监测本地的任务执行情况，计算最近一段时间内的任务执 行效率，即任务抓取速度。影响爬虫节点性能的因素很多，节点性能、网络状况 和目标网站的位置都会影响，但归根结底还是体现在爬虫整体的釆集速度上区］, 因此根据单位时间内抓取网页的数量，即任务执行效率来评价反馈任务分配。选 取的计算吋间段太长会使爬虫釆集速度趋于稳定，无法体现采集速度的变化，时 间段太短可能会因波动导致计算的采集速度与实际相差过大，通过观察，选取最 近3分钟进行计算得到的数值比较准确。

信息交互是指爬虫节点与中心存储集群中的节点状态存储单元进行信息交 换。更新本节点的状态信息，包含节点负载值和任务效率；获取系统内其他节点 的状态信息；获取任务信息。

动态调整与调度是爬虫节点根据自身负载与集群负载情况，结合自身任务效 率，计算得出需要领取的任务数量并获取任务。任务领取数量由评价系数加一后 与上次任务数量的乘积确定。评价系数的计算公式为式4-2。

q **=Vi\*（s-s,） + \*2\*（1F）+ V3\*（V/7；）/K** （4-2）

其中，；表示系统的平均负载值，祁表示爬虫当前釆集速度，也是上一次调 度后的采集速度，《表示爬虫上一次调度前的采集速度，乂、力和以为各项的 系数，均大于0。

式4-2中基于反馈的动态调整策略描述如下：

（1） 当系统中负载不均衡时，加大负载差值对评价系数的影响。当爬虫节

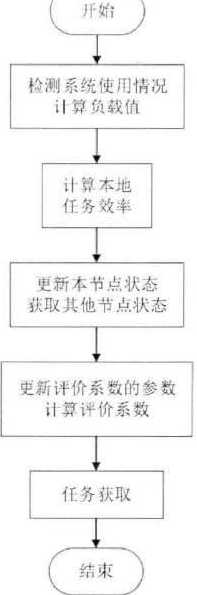
. 点的负载小于系统的平均负载时，式4-2的第一项为正值，将会对最终

评价值产生正的影响以增加其负载，反之将减小其负载。当节点间的负 载差异（；-s,）超过时，要增加负载在评价系数中的比重。当负载 严重不均衡发生时，每次任务调度将凹的值增加0.1,劣的值置为0,外 的值减小0.2,直到负载差异小于Z）”，随后恢复系数原值。

（2） 系统负载较为均衡的情况下，应该尽可能的提高系统利用率，即提高 系统的负载值。在系统负载差异小于Q『时，系统利用率越低，式4-2 的第二项对最终评价值的影响应该最大，才能以较快的速度提高任务分 配效率，最终提高系统整体负载率，提升效率。当系统负载差异小于*D.* 时，系统负载值每比80%低10%,力的值增加0.03o

（3） 爬虫下载速度越快，完成抓取任务的时间最少。任务的分配和节点的 负载会影响爬虫的下载速度，通过爬虫下载速度的变化反馈调整任务调 度，最终获取较快的下载速度。当系统负载差异小于。，若爬虫下载 速度增快，增速超过20%时\*3的值设为1，否则设为0.5；若爬虫下载 速度降低，比的值设为。・5。如果爬虫下载速度为0,不更新此的值。

任务调度的工作流程如图4-9所示。



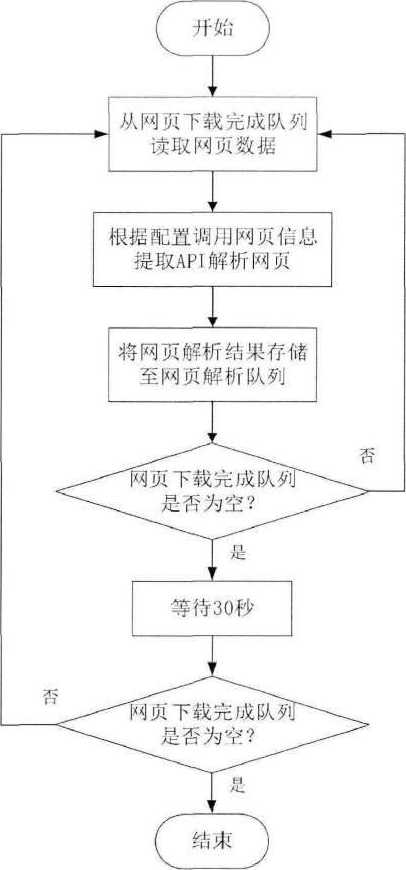
图**4-9**任务调度流程图

4.4.3网页信息提取模块的设计与实现

网页信息提取模块的主要功能是对网页下载器下载的网页源码进行解析，去 除网页中的冗余信息，提取有价值的信息。本系统釆用上文提出的网页信息自动 提取算法提取网页中的元数据。网页信息提取模块的工作流程如图4-10所示。

网页信息提取模块的详细流程具体说明如下：

1. 网页信息提取器从网页下载器的完成队列CompletedQueue读取键值 对为＜URL, webPageDTO＞的网页数据；
2. 调用网页信息提取API解析网页，网页信息提取API将在下文讲述；
3. 根据网页信息提取API的返回结果构建键值对＜URL,documentDTO＞, documentDTO包含的字段如表4-5所示，并将其放入解析队列 ParseQueue。
4. 判断网页下载完成队列是否为空，若为否，跳转至步骤(1),否则等 待30秒继续判断队列是否为空，仍为空则结束流程，不为空则跳转至 步骤(1)。等待一段时间二次判断是否为空的原因是可能会存在下载任 务尚未完成但下载完成队列为空的情况，30秒的时间足够网页下载器完 成一次网页下载。



图**4-10**网页信息提取模块流程图 表**4-5 documentDTO**包含的字段及其含义

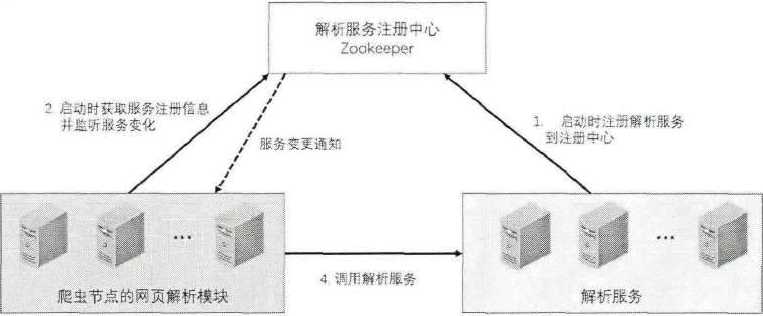
|  |  |  |
| --- | --- | --- |
| 字段 | 含义 | 类型 |
| htmlSource | 网页源码 | String |
| crawl Time | 抓取时间 | DateTime |
| crawlerName | 爬虫节点 | String |
| parseTime | 解析时间 | DateTime |
| parserName | 解析器 | String |
| articleContent | 文章内容 | Json |

上文提到网页信息提取实际在网页信息提取服务提供的API中完成，将网 页信息提取服务封装成RESTfiil风格的API,以微服务的形式对外提供网页信息 提取能力，这样做的好处是网页信息提取服务与爬虫服务通过低耦合的方式交互,

便于网页信息提取服务的迭代升级与扩展；当分布式爬虫系统扩容，需要更强的 网页信息提取能力时便于对网页信息提取服务进行水平扩展。

下面叙述网页信息提取服务的设计与实现，网页信息提取服务的架构图如图

4-11所示。



图**4-11**网页信息提取服务架构图

网页信息提取服务主要涉及三个部分：

（1） 解析服务

解析服务的提供方，由部署了网页信息自动提取算法的多个服务器组成。 使用Python中的轻量级web框架Flask将网页信息自动提取算法封装为 API,使用Nignx（Enginex）web代理服务器将应用服务部署在物理服务 器上对外提供服务。应用服务在启动时注册解析服务到注册中心，在注 册中心创建一个服务节点并存储应用的服务名称和服务地址等相关数 据。应用服务的流程图如图竺婴。

（开始）

加载网页信息自动

提取算法

—丄

向服务注册中心

注册

Jz

等待接口调用（一  
丄 二

网页信息提取——

图**4-12**解析服务流程图

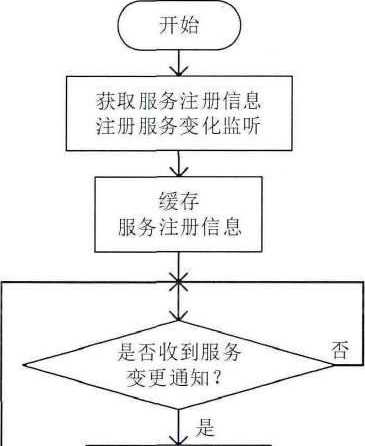
（2） 解析服务注册中心

解析服务注册中心使用Zookeeper,用于存储解析服务提供者的相关数

据，维护服务注册表。当解析服务上线或下线(如服务提供者对服务进 行维护升级或不再提供服务)时会向爬虫节点发送通知，服务下线的感 知通过心跳检测机制实现。

(3)爬虫节点的网页信息提取模块

网页信息提取模块在启动时通过服务注册中心发现已注册的解析服务 并将解析服务信息缓存到本地，同时监听服务变化，当收到服务变更通 知时更新本地缓存。在网页信息提取时随机选择某台服务器的解析服务 进行调用。网页信息提取模块服务发现相关的流程图如图4-13所示。



更新

服务注册信息缓存

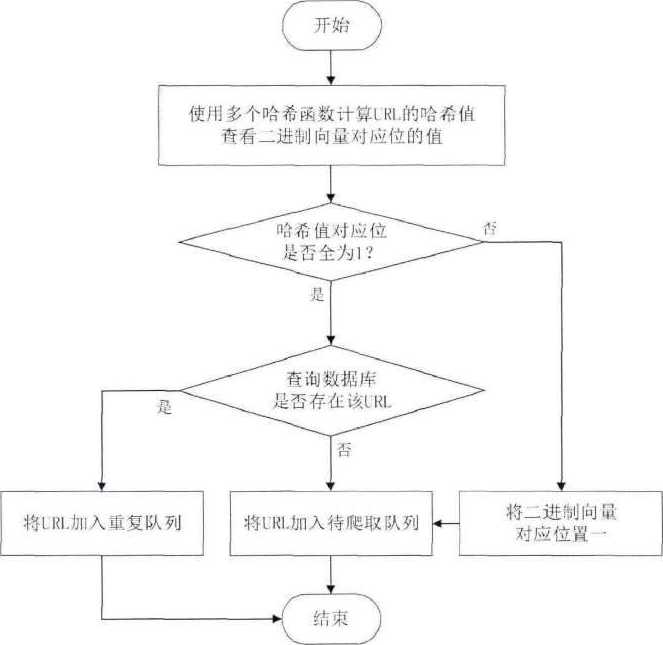
图**4-13**节点服务发现流程图

4.4.4网页去重模块的设计与实现

网页去重模块实现对网页的去重，分为URL去重和网页内容去重。URL去 重的目的是防止对相同的URL重新爬取，提高爬虫的爬取效率⑵。网页内容去 重针对重复网页和内容高度相似网页，这些网页在主体部分内容基本相同，调査 研究发现，因特网上的近似网页数量达到了网页总数的30%〜45%左右网。解决 数据重复、冗余的问题是数据处理分析的前提条件。下面阐述如何设计与实现 URL去重和网页内容去重。

URL去重检查已下载的URL中是否存在该记录，如果存在则移除该URL。 目前有如下几种URL重复判断方法：基于数据库去重、基于Set去重、基于Trie 树去重和基于Bloom Filter的去重，下面是几种方法的对比：基于数据库去重效 率低，因为数据库读数据比较慢；基于Set去重效率高，但空间复杂度同样比较 高，而且Set去重采用哈希函数，可能因为哈希碰撞导致判断错误；基于Trie树 去重效率很高，但数据结构复杂，URL由ASCII码组成的，所以用于URL去重 的Trie树是128叉树，而且部分URL长度较大，Trie树的深度可能超过100； 基于Bloom Filter的去重具有最优的时间和空间效率，唯一的缺点是有一定的误 判率，多个映射函数可以降低碰撞概率，碰撞概率极低在海量数据的场景下也可 能会出现碰撞。

考虑到分布式爬虫系统中海量URL的场景，本文釆用基于Bloom Filter的 去重实现URL去重以获取最优的时间和空间效率，针对可能会岀现误判这一情 况，本文在工程上解决这一问题：当出现碰撞时，通过数据库査询己下载网页中 是否存在该URL做出是否重复判断。URL去重流程如图4-14所示。



图**4-14 URL**去重流程图

网页内容去重是在网页下载解析完成之后，对提取出的网页内容计算网页指 纹，将网页间指纹的相似度或距离看作网页相似度。只对提取出网页内容计算相 似度可以避免网页噪声的影响，从而提高网页去重的准确率。

实现网页内容去重的关键有两点：一是如何计算网页指纹，将原始高维的数 据映射到低维的数据空间中，并且相邻的数据点在映射后依然保持相邻或邻近㈣】, 谷歌提出的局部敏感哈希(Locality Sensitive Hash, LSH)算法可以解决该问题⑹｝。 二是网页相似度判断。网页数量很大时，如何以较快的速度检查新网页与己有网 页的指纹相似度，如果己有网页的数量为100万，遍历比较就要进行100万次，

过大的比较耗时是无法接受的，本文利用“抽屉原理”采取空间换时间的方案解 决这个问题。

(1)网页指纹计算

Charikar提出的Simhash算法是常见的LSH算法，本文使用Simhash算法计 算网页指纹，指纹生成过程如图4-15所示。

(feature, weight)

(hash, weight)



(tl. wl)

(t2. w2)

Hash

(100110, wl)

(110000. w2)

加权

wl -wl -wl wl wl -wl w2 w2 -w2 -w2 -w2 -w2

(tn. wn)

(001001, wn)

wn wn -wn wn wn - wn

Add

指纹

110001

12 156 -45 -15 -78496

图4-15 Simhash算法指纹生成示意图

Simhash算法的网页指纹计算过程如下：

1. 分词。对网页文本进行分词，同时设置每一个分词的权重，分词的权重 weight使用文本频率与逆文档频率指数(Term Frequency - Inverse Document Frequency, TF-IDF)获取。本文使用jieba分词器进行分词。
2. Hash o通过Hash函数计算各个特征向量的hash值。本文使用 MurmurHash函数生成64bit的哈希值。
3. 加权。在Hash值的基础上，对所有特征向量进行加权。如果Hash值为

“10110011……00111100”,权重为3,那么加权后的序列为“3-3 3 3

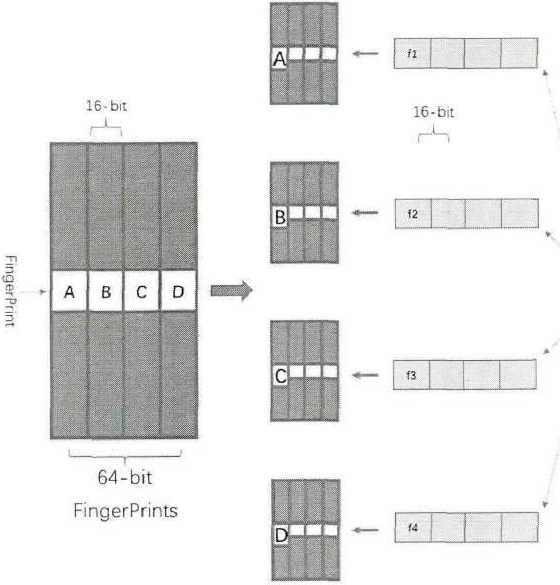
-3 -3 3 3 -3 -3 3 3 3 3 -3 -3”。

1. 合并。累加加权后的各个特征向量。
2. 降维。对上一步的结果序列串降维，大于0则置1,否则置0,最终得 到64位二进制指纹。

(2)网页相似度判断

通过上一步，得到了网页的64位二进制指纹，网页指纹间的汉明距离可以 表征网页的相似度，汉明距离越小，网页内容越相似。经验表明，对于64位的 网页指纹，汉明距离在3及以下的网页可以认为是重复网页。

在确定网页相似度判断的标准之后，接下来需要在己存储的海量网页中找到 相似网页，与已有的网页指纹逐一计算汉明距离是不现实的，因为这会耗费大量 时间。对海量数据的查询或者检索通常使用建立索引的方式加快查找速度，使用 索引可以减少比较次数。HashMap是一种理论上具有0(1)时间复杂度的查找数 据结构，网页指纹如果能够利用HashMap建立索引无疑能够显著的减少比较时 间。利用“抽屉原理”，将64位的网页指纹拆分为4个16位的指纹，如果两个 网页的海明距离在3以内，那么至少有1个16位指纹完全相同。由此，可以通 过4个16位的指纹使用HashMap建立倒排索引,釆用空间换时间的方式加快比 较速度，索引建立的示意图如图4-16所示。在命中索引后，对索引后的链表数



据按顺序进行指纹比较。

图**4-16**网页指纹建立倒排索引示意图

**new 64-bit Fingerprint**

如果网页库中有2人25个网页，每新增一个网页，建立索引前需要进行2人25 次汉明距离计算，建立索引之后只需要进行2^25-16)=512次比较，建立索引大 大减少了汉明距离的计算成本。建立4个倒排索引后的存储空间是一个64位索 引空间的4倍，2&5个指纹需要256MB,扩大4倍为1GB,存储代价可以接受。

本文实现的基于Simhash算法的相似网页査找算法如表4-6所示。

表**4-6**基于**Simhash #**法的相似网页查找算法

**算法：基于Simhash的相似网页査找算法**

输入：网页内容文档

输出：重复网页指纹集合

**FindSimilarWebPages:**

01:初始化64维空间向量Sum,初始化重复网页指纹集合R

02:分割提取后的网页信息的特征词，记录权重Wi

03: while特征词集合F不为空

04: 通过Hash函数Murmur计算特征词Fi的哈希值h；

05: 将哈希值h映射成64维空间向量bi,空间向量bi的各个分量进行

加权，将空间向量的各个分量与权重相乘，如果分量为1,该位则为权 重Wi,如果为0,该位则为权重负值，即-Wi；

06: 将空间向量bi与空间向量Sum相加，将和赋值给Sum；

07: 将Fi从特征词集合F删除；

08: End while

09:对空间向量Sum降维，各分量大于0则置1,否则置0,降维后将其转

化为二进制字符串，得到网页指纹玲

10:将网页指纹切分为4个，通过索引将命中数据加入疑似重复集合Fs

11: While集合Fs不为空

12: 计算网页指纹坊与指纹Fsi的汉明距离d

13: If d小于等于3 then

14: 将指纹Fsi加入重复网页指纹集合R

15: End if

16: 将Fsi从集合Fs删除

17: End while

18: Return重复网页指纹集合R

4.4.5节点管理模块的设计与实现

节点管理模块包含节点的接入管理和节点状态监控及告警。

节点的接入管理是动态管理节点加入和退出分布式系统。节点加入集群分为 新增节点和异常节点复活，节点退出集群是系统管理员使节点休眠并删除节点。 节点的状态分为休眠、活跃、异常、已删除四种，状态转移图如图4-17所示。 节点加入集群后状态为休眠，此时节点存活但不执行抓取任务，管理员将节点激 活为活跃状态后执行抓取任务，当节点失效后状态转变为异常，节点是否失效由集群中其他节点共同判断，异常节点为失活节点，管理员可将休眠或异常节点删 除。

“点加丿字理员激活/休眠节点

管理员，删除

异常恢复

,

节点失效

管理员删除异常节点

异常

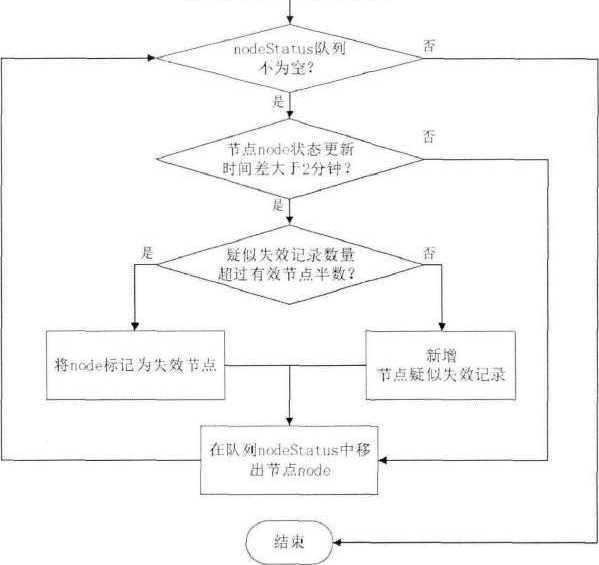
S

图**4-17**节点状态转移图

节点通过将自身信息加入中心存储集群中节点信息共享存储单元的方式加 入集群并定时更新自身的状态信息，其他节点通过对节点信息共享存储单元的定 时监测发现新节点，各节点在节点信息共享存储单元得到的节点信息是一致的， 能够保证分布式系统数据的最终一致性。节点的异常恢复在异常情况处置之后与 节点新增的加入方式相同，区别在于新增的节点初始状态为休眠，恢复的节点初 始状态为活跃。

更新自身状态,  
淸空节点疑似失效记亲  
vot di ecords

获取其他节点状态信息. 组成队列nr»dcStatus



图**4-18**节点状态更新流程图

节点失效由集群中的其他节点共同判断，上文提到集群中的每个节点会定时 更新自身的状态信息，同时会检查每个节点的信息新鲜度，即当前时间和该节点 上次状态更新的时间差，如果时间差大于2分钟，说明该节点可能出现异常，检 査节点状态的voteRecords字段中的信息，如果不同节点的疑似失效记录超过当 前有效节点数量的一半，则将该节点标记为失效节点，否则在voteRecords中增 加疑似失效记录。节点每次更新状态会将自己的voteRecords清空。节点状态更 新流程图如图4-18所示。

节点状态监控分为爬虫节点运行信息监控和节点服务器状态监控，分别监控 爬虫节点的运行信息和爬虫节点所在服务器的状态，监测的各项指标如表4-7和 表4-8所示。

表4-7爬虫节点运行信息

|  |  |
| --- | --- |
| 统计指标 | 指标含义 |
| spiderName | 爬虫名称 |
| startTime | 爬虫启动时间 |
| spiderStatus | 爬虫状态。正在运行：running；停止： stopped； 异常：abnormal |
| averageCrawl Speed | 网页抓取平均速度 |
| downloadCount | 成功下载的网页数量 |
| downloadF ai lureCount | 下载失败的网页数量 |
| parseCount | 爬虫成功解析的网页数量 |
| parseF ai lureCount | 爬虫解析失败的网页数量 |
| storeCount | 爬虫成功存储到数据库的网页数量 |
| storeFailureC ount | 爬虫数据库存储失败的网页数量 |

表4-8节点状态

|  |  |
| --- | --- |
| 节点状态指标 | 指标含义 |
| CPUUsageRate | CPU使用率 |
| networkSpeed | 当前网速 |
| memoryU sageRat | 消耗的内存 |
| diskUsageRat | 当前磁盘使用率 |

由于爬虫的运行环境有着诸多的不确定性〔62〕,爬虫的运行过程中可能会遇到 异常。异常最终都会反映到爬虫节点的运行状态和服务器状态。为了保证异常能 够被及时处理从而保证顺利完成抓取任务，在节点状态监控的基础上，本文设计 了告警的功能，当出现异常时，根据告警配置及时通过短信和邮件的方式对系统

使用者进行示警。

根据异常的类型及重要程度，将告警分为三个等级：warning, error和fatal, 异常类型及告警等级如表4-9所示。

表4-9异常类型及告警等级表

|  |  |  |
| --- | --- | --- |
| 告警等级 | 异常类型 | 含义 |
| fatal | serverLosl | 节点服务器失去连接 |
| error | downloadError | 网页下载错误 |
| error | parseError | 网页信息提取错误 |
| error | storageError | 网页存储错误 |
| error | fbrbiddenError | 爬虫被禁止访问错误 |
| warning | overloadWaming | 节点负载过重警告 |
| warning | downloadWaming | 网页下载失败警告 |
| warning | parseWaming | 网页信息提取失败警告 |
| warning | storage Warning | 网页存储失败警告 |

其中，网页的下载、解析和存储都有warning和error等级的告警，告警等 级根据单位时间内失败的URL数量或者一天内失败的总数量来区分，偶然性的 错误为warning,持续性的错误为error,需要设置合理的告警阈值，默认3分钟 内下载、解析或存储失败超过10次以及一天内失败超过50次为error,用户可 根据实际情况自行配置。告警配置页面如图4-19所示。

图**4-19**告警页面配置图 当节点出现异常时，自动发送告警邮件和告警短信，告警邮件的模板如下： 爬虫运行异常，请及时处理，异常信息如下： 爬虫名称：｛ spiderName ｝, 异常级别：｛ exceptionLevel ), 异常类型：｛ exceptionType｝, 告警时间：｛ alertTime ｝, 详情：｛ exceptionDetail ｝

短信告警模板如下：

爬虫｛spiderName ｝出现异常｛exceptionType ｝,请及时査看处理，

｛ alertTime )

4.4.6爬虫管理模块的设计与实现

爬虫管理模块主要实现爬虫任务的配置和管理功能。

爬虫任务配置主要包含以下项：名称、任务描述、是否为周期任务及间隔时 间、是否使用IP代理、是否启用下载间隔及间隔时间、URL集合和任务优先级 等。任务配置如图4-20所示。

、走勞

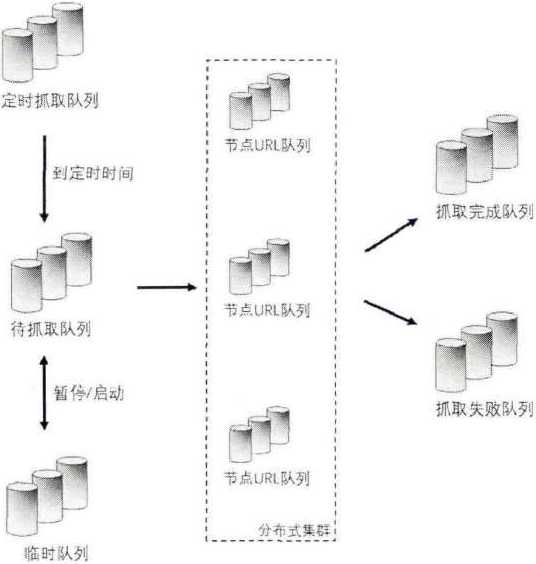
.会笛

|  |  |
| --- | --- |
|  |  |
| AFX |  |
| 冬莎 |  |
|  | 蚱与7•'曜 |
|  |  |
|  | 豪 |
|  | ：•? t\* t\* |
|  | 3, W\*，、 ES.-' |
|  | —C Ci X 8 w |

图4.20爬虫任务配置图

爬虫任务的管理是对抓取任务的启动、停止与删除，包含周期性任务的定时 启动，通过对URL的处理来实现。爬虫任务管理中的URL流转如图4-21所示。 爬虫任务配置完成后URL首先进入初始队列，爬虫任务被启动后URL将进入待 抓取队列等待爬虫节点抓取，任务被暂停后对应的URL从待抓取队列移入临时 队列，周期性任务启动后生成URL到定时抓取队列，到定时时间后从定时抓取

队列移入待抓取队列。



初始队列

图**4-21**任务管理**URL**流转图

周期性任务的抓取通过周期性的将URL加入到待抓取队列实现，流程如图 4-22所示：

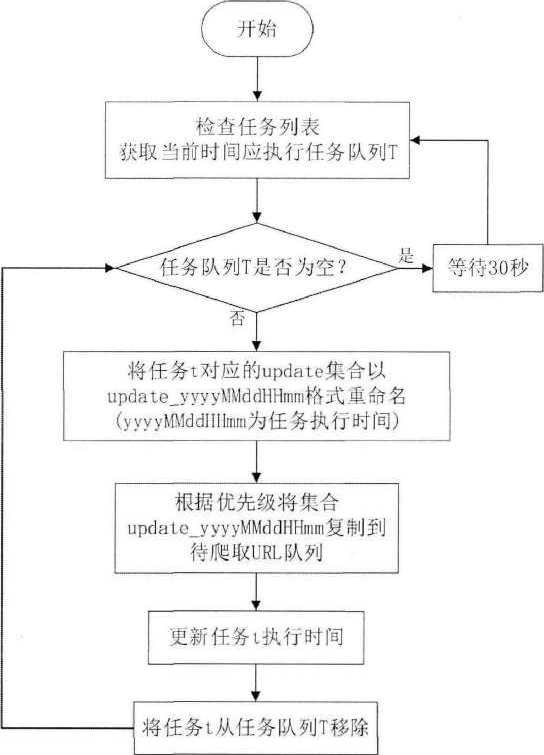
（1） 检查任务列表，获取当前时间应执行任务队列，如果没有应执行任务， 继续等待30秒再次检查任务列表；

（2） 遍历应执行任务队列，将应执行任务在定时抓取队列中对应的update 集合以“update\_yyyyMMddHHmm ”格式重命名,其中“yyyyMMddHHmm " 为任务执行时间；

（3） 根据任务的优先级将update\_yyyyMMddHHmm集合复制到待爬取队 列中的不同子队列，等待爬虫节点读取；

（4） 在任务列表更新下一次任务执行时间。

周期性任务的抓取周期通常为1天，可根据实际情况将爬虫任务的起始时间 分散在一天中的不同时段。



图**4-22**定时任务流程图

4.5系统测试

本小节对上文实现的分布式爬虫系统进行测试，包括系统功能验证和性能测 试。系统的测试基于2台运行CentOS7的物理主机，每台物理主机的配置为： Intel(R)Xeon(R)@2.4GHz 8 核 16 线程 CPU、16G 内存以及 500G 硬盘，100M/ 秒的带宽，将每台主机配置为4台性能相同的虚拟主机，分别记为虚拟主机A-H, 其中主机A-D属于同一台物理主机。

4.5.1功能测试

首先对本系统的每个功能进行验证，测试方案及结果如表4-10所示。

表4-10功能测试方案及结果

|  |  |  |
| --- | --- | --- |
| 测试功能 | 测试方案 | 测试结果 |
| 网页下载 | 选取100个网页URL链接，设置爬取间隔1 秒，打印下载的网页信息及下载时间 | 网页下载功 能正常 |
| 网页信息提取 | 选取100个页面打印URL和各项网页元数据 | 网页信息提 取功能正常 |
| 网页去重 | 1. 选取3组重复网页URL链接，每组20个， 打印实际下载的网页URL； 2. 选取50个网页URL链接，包含3个内容相 似网页（转载的新闻或博客），打印网页指纹 | URL去重和 内容去重功 能正常 |
| 网页存储 | 选取100个网页URL链接，抓取后査看数据 库网页记录 | 网页存储符 合预期，功能 正常 |
| 任务调度 | 选取1000个网页URL链接，分别使用2个爬 虫节点和4个爬虫节点抓取，打印节点状态和 下载的网页URL | 任务调度符 合预期，功能 正常 |
| 节点管理 | 爬虫系统执行抓取任务时新增节点和休眠节 点，打印所有节点状态信息 | 节点管理功 能正常 |
| 爬虫任务管理 | 启动、暂停和删除爬虫任务，查看任务状态及 节点日志 | 任务功能正 常 |
| 系统监控 | 爬虫系统执行任务时关闭节点网络连接或关 闭节点，査看是否收到邮件或短信通知 | 监控告警功 能正常 |

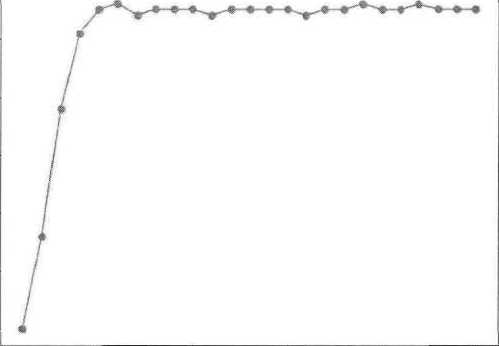
4.5.2性能测试

本小节对系统进行性能测试，分为系统稳定性测试、抓取速度测试和系统横 向扩展性测试。

本文选取5000个网页逬行稳定性测试，将下载间隔设置为10秒以使分布式 爬虫系统长时间工作，在测试开始的一段时间内，各个节点正常工作，节点负载 逐渐趋于稳定且无较大波动，任务调度正常，数据存储正常；在系统运行1小时 后断开主机A的网络连接，收到告警邮件和短信，此时系统任务调度正常，数 据存储正常。在整个测试过程中网页下载与解析正常，数据存储正常，测试结果 符合预期。査看数据库中的数据，5000个网页全部有数据记录，网页信息提取 的数据结果符合抓取需求，达到了 97%的爬全率。

本文选取5000个网页，使用单个虚拟机节点进行抓取速度测试，不设置下

载间隔且使用1P代理，抓取速度如图4-23所示。随着运行时间的增加，抓取速 度逐渐加快最终达到一个稳定的值。



**10 15**

**timezone (minute)**

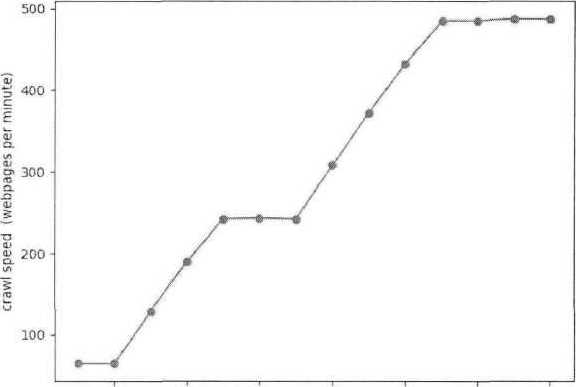
**20**

50403020

(I'au-ul Ed sgedq\*,) p$ds 一\*二

图**4-23**单节点爬虫系统抓取速度

横向扩展性是衡量分布式系统性能的关键指标之一，横向扩展性好的分布式 系统增加节点即可增加系统的总体性能。本文使用8个虚拟主机进行横向扩展性 测试，测试开始时系统只有虚拟主机A一个节点，十分钟后每隔5分钟加入一 台虚拟主机节点，其中虚拟主机D加入集群后多等待5分钟再继续加入其它节 点。每隔5分钟记录一次系统整体抓取速度，抓取速度变化如图4-24所示。



10 20 30 40 50 60 70

timezone (minute)

图**4-24**多节点爬虫系统抓取速度

由图4-24可知，当节点增加时，网页抓取速度也会增加，当节点数量较少 时，增加性能相同的节点抓取速度近乎线性增长。同时注意到同一台物理主机对 应的虚拟主机依次加入集群时抓取速度的增加逐渐变慢，分析发现是由于物理主 机的总带宽有限，虚拟机管理软件不能严格的限制每台虚拟的带宽。如果将每台 物理主机作为系统中的一个节点，同样可以得岀增加性能相同的节点时抓取速度 近乎线性增长。

4.6本章小结

在网页信息自动提取算法的基础上，本章设计实现了分布式网络爬虫系统。 本章首先结合企业实际项目需要进行了需求分析，然后对系统进行总体设计，将 系统分为存储、网页下载与任务调度、网页信息提取、网页去重、节点管理和爬 虫管理六个模块，根据模块的功能设计和实现了各个模块。最后，为了确保系统 能够满足企业需求，对系统进行了功能和性能测试。

第五章总结与展望

5.1论文工作总结

互联网每天产生海量的数据，这些数据蕴含的信息具有巨大的研究价值和商 业价值。学者及企业希望髙效的获取资讯类、知识类、政策法规类等文章型网页 中最新有价值的信息。现实中网页结构各异且都包含大量网页噪声，本文主要研 究提出了一种基于视觉块一致性和序列标注的文章型网页元数据提取算法并设 计实现了基于此算法的分布式网络爬虫系统，能够在海量信息中以较高的效率获 取有价值的结构化信息。本文的主要工作分为以下三个方面：

1. 相关技术综述。本文首先对网络爬虫技术的现状以及应用进行了综述； 接下来对网页信息提取技术进行了概述，为算法研究奠定基础；最后梳理了分布 式系统的相关技术和相关研究，为分布式系统的设计奠定基础。
2. 针对现有文章型网页信息提取算法提取精度不足、提取信息缺失和上 下文信息利用较少的问题，提出了一种文章型网页信息自动提取算法。首先，本 文对文章型网页的数据模型做了介绍，然后从视觉特征、文本特征和DOM树特 征三个方面对文章型网页进行了分析，接下来提出了基于视觉块一致性和序列标 注的文章型网页元数据提取算法，即利用视觉特征将DOM树节点划分为多个一 致性视觉块，根据视觉块的视觉特征和文本特征定位网页主体区域，使用主体区 域视觉块的文本特征、视觉特征和词典特征训练CRF模型，识别文章标题、作 者、发布时间、摘要、正文、附件、图片等网页元数据，完成网页信息的自动提 取，最后对该算法的提取效果进行了实验验证。
3. 分布式爬虫系统的设计与实现。首先，本文通过分析企业的需求确定 系统的基本功能，然后对系统进行总体设计，从下至上的将系统分为数据釆集层、 数据解析层、数据存储层、节点接入层和系统管理层，各层共包含存储、网页下 载与任务调度、网页信息提取、网页去重、节点管理和爬虫管理等六个模块，根 据模块的功能设计和实现了各个模块。最后，本文对系统进行了功能和性能测试。

**5.2**下一步工作展望

本文研究提出了一种文章型网页信息自动提取算法并设计实现了分布式网 络爬虫系统，为大规模文章型网页的信息获取提供可行的工具与方案。实验结果 表明，该算法及系统取得了不错的效果，但仍存在一些可以继续深入研究的方向， 主要包含以下几个方面：

1. 本文研究对文章型网页的信息提取，互联网中还存在大量的评论型和 列表型网页，这些网页同样蕴含着巨大的研究价值和商业价值。如评论型网页中 的电子商务产品评论数据和博客评论数据，利用产品评论，企业可以洞察产品优 劣和用户喜好，从而可以改善商业决策；利用博客评论数据可以追踪分析不同话 题的趋势，进行舆情监测。由于这些网页中包含的信息具有巨大的价值，网站服 务提供者设置了完善的反爬取措施，如何应对反爬取措施实现评论型和列表型网 页的信息提取是当前的一个难点。在未来的工作中，本文将继续研究评论型和列 表型网页的信息自动提取算法，使分布式爬虫系统能够抓取互联网中绝大部分网 页并提取其信息。
2. 现阶段本文使用无头浏览器进行网页渲染以获取网页的视觉信息，渲 染过程会耗费大量时间，占据了一个网页下载解析的大部分时间，在网页数量急 剧增长的背景下降低系统整体的的时间复杂度非常重要，优化网页视觉信息的获 取过程以降低系统的时间复杂度也将成为未来的一个研究点。

参考文献

1. .中国互联网信息中心.中国互联网络发展状况统计报告[EB/OL].

<http://www.cac.gov.cn/2020-09/29/c_l602939918747816.htm>, 2020.

1. .荣哈.基于分布式的网络爬虫系统的研究与实现[D].电子科技大学,2017.
2. .李优.基于Hadoop的分布式网络爬虫设计与实现[D].西北大学,2018.
3. .杜凤媛.自定义分布式网络爬虫的设计与实现[D].电子科技大学,2019.
4. .谢方立.基于节点类型标注的网页主题信息提取技术研究[D].中国农业科学

院,2016.

1. .耿令宝.分布式环境下的网络爬虫系统研究与优化[D].北京邮电大学,2015.
2. .刘星辰.基于Hadoop的分布式网络爬虫的研究与实现[D].西安理工大

学,2019.

1. .孙立伟，何国辉，吴礼发.网络爬虫技术的研究[J].电脑知识与技

术,2010,6(15):4112-4115.

1. . Chakrabarti S, Van den Berg M, Dom B. Focused crawling: a new approach to

topic-specific Web resource discovery[J]. Computer networks, 2009, 3)(11-16): 1623-1640.

1. , Cho J, Garcia-Molina H. The evolution of the web and implications for an incremental crawler[R]. Stanford, 1999.
2. . Cho J, Garcia-Molina H. Synchronizing a database to improve freshness[J]. ACM sigmod record, 2010, 29(2): 117-128.
3. .马联帅.基于Scrapy的分布式网络新闻抓取系统设计与实现[D].西安电子 科技大学,2015.
4. .马倩.基于下一代web技术的移动网页模板匹配的设计与实现[D]・北京邮 电大学,2016.
5. . Wang J, Lochovsky F H. Data-rich section extraction from html pages[C]//Proceedings of the Third International Conference on Web Information Systems Engineering, 2002. WISE 2002. IEEE, 2002: 313-322.
6. , Bar-Yossef Z, R^agopalan S. Template detection via data mining and its applications[C]//Proceedings of the 11th international conference on World Wide Web. 2002: 580-591.
7. ,王宪发，郭岩，刘悦，俞晓明,程学旗.基于视觉特征的网页信息抽取方法研究

卩],中文信息学报,2019,33(05):103-112.

1. .高峰.WEB实体信息的提取算法及其应用研究[D].电子科技大学,2018.
2. .孙承杰，关毅.基于统计的网页正文信息抽取方法的研究卩].中文信息学 报,2004(05):17-22.
3. .韩忠明，李文正，莫倩.有效HTML文本信息抽取方法的研究卩].计算机应用研 *究*,2008,25(12):3568-3571 +3574.
4. . Song M, Wu X. Content extraction from web pages based on Chinese punctuation number[C]//2007 International Conference on Wireless Communications, Networking and Mobile Computing. IEEE, 2007: 5573-5575.
5. .王琦，唐世渭,杨冬青，王腾蛟.基于D0M的网页主题信息自动提取[A].中 国计算机学会数据库专业委员会.第二十一届中国数据库学术会议论文集(研 究报告篇)[C].中国计算机学会数据库专业委员会:中国计算机学会数据库专 业委员会,2004:7.
6. ,王少康，董科军,阎保平.使用特征文本密度的网页正文提取[J].计算机工程与 应用,2010,46(20):1-3+29.
7. .高庆宁，吴鹏，张晶晶.基于文档对象模型与行块分布算法的网页信息抽取卩]. 情报理论与实践,2016,39(04):133-137.
8. .李文昊，彭红超，童名文,石俊杰.基于视觉特征的网页最优分割算法[J].计算 机科学,2015,42(11):284-287+309.
9. , Cai D, Yu S, Wen J R, et al. Vips: a vision-based page segmentation algorithm [J]. 2003.
10. ,孙景春.基于视觉块识别的网页元数据提取方法[D].东南大学,2017.
11. .何正华，顾君忠，杨静.基于视觉特征的BBS用户发言信息抽取[J].计算机应 用,2009,29(81):171-174.
12. .吴倩，杨逍，张兆心.基于视觉特征的网页信息提取[A].中国中文信息学会 信息检索与内容安全专业委员会.第六届全国信息检索学术会议论文集[C].中 国中文信息学会信息检索与内容安全专业委员会:中国中文信息学会,2010:8.
13. . Michael Cormier, Karyn Moffatt, Robin Cohen, et al. Purely vision-based segmentation of web pages for assistive technology. 2016, 148:46-66.
14. .朱冰阳.基于CEF的页面分割算法的研究[D].中国海洋大学,2015.
15. .安增文，王超，徐杰锋.基于机器学习的网页正文提取方法[J].微型机与应 用, 2010(12): 4-6.
16. .吕容政，刘嘉勇.基于决策树的自适应网页正文抽取方法[J].现代计算机(专业

版),2019(07):16-22.

1. . Debnath S, Mitra P, Pal N, et al. Automatic identification of informative sections of web pages[J], IEEE transactions on knowledge and data engineering, 2005, 17(9): 1233-1246.
2. . Wu S, Liu J, Fan J. Automatic web content extraction by combination of learning and grouping[C]//Proceedings of the 24th international conference on World Wide Web. 2015: 1264-1274.
3. .余科军.分布式实时系统任务调度算法的设计和实现[D].四川大学,2006.
4. .王霓虹，张露露.分布式爬虫任务调度策略的优化[J].黑龙江大学自然科学学 报,2016,33(05):671-675+701.
5. .刘林东.分布式异构环境中任务调度算法研究[D].华南理工大学,2019.
6. . Liu L, Qi D, Zhou N, et al. A task scheduling algorithm based on classification mining in fog computing environment[J]. Wireless Communications and Mobile Computing, 2018, 2018.
7. . Fan Y, Tao L, Chen J. Associated task scheduling based on dynamic finish time prediction for cloud computing[C]//2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2019: 2005-2014.
8. .房婷.异构分布式环境下任务调度问题的研究[D].大连理工大学,2014.
9. .区基伟.基于页面模块识别的Web文章自动抽取方法研究[D].哈尔滨工业 大学,2012.
10. .穆琼.基于视觉特征的网页清洗研究与实现[D].北京邮电大学,2014.
11. ,周艳平，李金鹏,宋群豹.一种基于SVM及文本密度特征的网页信息提取方法 [J].计算机应用与软件,2019,36(10):251-255+261.
12. ,邵振凯.网页信息提取技术[J].计算机技术与发展,2013,23(09):36-38+42.
13. , Song X, Liu J, Cao Y, et al. Automatic extraction of web data records containing user-generated content[C]//Proceedings of the 19th ACM international conference on Information and knowledge management. 2010: 39-48.
14. .肖悦.基于文本密度和页面结构的网页信息抽取技术研究与实现[D].中国 海洋大学,2015.
15. .薛欢欢.基于条件随机场的中文期刊论文信息识别与抽取[D].中国农业科 学院,2019.
16. . Zhu J, Nie Z, Wen J R, et al. 2d conditional random fields for web information extraction[C]//Proceedings of the 22nd international conference on Machine learning. 2005: 1044-1051.
17. .于洪涛，虞海明,张付志.基于三阶条件随机场的论文元数据提取方法[J].小型

微型计算机系统,2014,35(03):606-609.

1. . Freitag D. Information extraction from HTML: Application of a general machine learning approach[C]//AAAI/IAAI. 2008: 517-523.
2. .向菁菁，耿光刚,李晓东.一种新闻网页关键信息的提取算法[J].计算机应 用,2016,36(08):2082-2086+2120.
3. . Sarr E N, Ousmane S, Diallo A. FactExtract: automatic collection and aggregation of articles and journalistic factual claims from online newspaper[C]//2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS). IEEE, 2018: 336-341.
4. ,朱少民，韩莹.软件项目管理[M].人民邮电出版社.2015.
5. ,汪兵.基于Scrapy框架的分布式爬虫系统设计与实现[D].合肥工业大 学,2019.
6. .刘艳红.网络爬虫行为的刑事规制研究一一以侵犯公民个人信息犯罪为视 角卩].政治与法律,2019(11):16-29.
7. ,章钺.基于Redis的分布式缓存系统的设计与实现[D].东南大学,2018.
8. ,孙瑜.基于Scrapy框架的网络爬虫系统的设计与实现[D].北京交通大 学,2019.
9. .樊宇豪.基于Scrapy的分布式网络爬虫系统设计与实现[D].电子科技大 学,2018.
10. .曹玉娟，牛振东，赵望，彭学平.基于概念和语义网络的近似网页检测算法卩丄 软件学报,2011,22(08): 1816-1826.
11. .陈杰.基于主题的海量中文文档去重技术研究[D].北京工业大学,2017.
12. . Oliver J, Cheng C, Chen Y. TLSH—a locality sensitive hash[C]//2013 Fourth Cybercrime and Trustworthy Computing Workshop. IEEE, 2013:7-13.
13. .李笑语.深度可定制的工具化爬虫系统的设计与实现[D].北京邮电大 学,2018.

缩略语

|  |  |  |
| --- | --- | --- |
| 英文缩写 | 英文全拼 | 中文全称 |
| API | Application Programming Interface | 应用程序接口 |
| CRF | Conditional Random Field | 条件随机场 |
| DAG | Direct Acyclic Graph | 有向无环图 |
| DOM | Document Object Model | 文档对象模型 |
| HTML | HyperText Markup Language | 超文本标记语言 |
| HTTP | HyperText Transfer Protocol | 超文本传输协议 |
| ID | Identity Document | 身份标识 |
| JDBC | Java Database Connectivity | Java数据库连接 |
| LSH | Locality Sensitive Hash | 局部敏感哈希 |
| MVC | Model View Controller | 模型视图控制器 |
| RDS | Relational Database Service | 关系型数据库 |
| SOA | Service-Oriented Architecture | 面向服务架构 |
| SQL | Structured Queiy Language | 结构化查询语言 |
| TF-1DF | Term Frequency-Inverse Document | 文本频率与逆文档频率 |
|  | Frequency | 指数 |
| URL | Uniform resource locator | 统一资源定位符 |
| Xpath | XML Path Language | XML路径语言 |

致谢

在北京邮电大学三年的硏究生生活中，长期得到各位老师和同学的关心与支 持，我在学术研究和工程实践方面的能力都有了明显提升，在这里向他们表示深 深的谢意。

首先，我要感谢我的导师夏海轮老师，夏老师在学习和生活上给予了我很大 的帮助。夏老师为人谦逊、严谨治学，工作认真精益求精，生活乐观积极向上， 在学习、工作和生活上都给我们树立了良好的榜样。不管是学习还是企业项目中 遇到的困难，夏老师总是乐于解答，循循善诱，帮助我解决了很多困惑和不解。 在此，向夏老师表示最诚挚的敬意和感谢！

其次，感谢实验室曾志民老师、冯春燕老师、郭彩丽老师、张天魁老师和刘 芳芳老师在我的整个论文工作阶段给予的指导和帮助。各位老师对学术论文严谨 和认真的态度使我在做学术研究及项目开发的过程中受益匪浅，让我在专业方面 的知识得到了较深的理解，在专业技能方面也得到了较大的提高。

此外，感谢中国电信集团系统集成有限责任公司的领导们和同事们，在工作 期间学到了各种知识，收获了友情，感谢你们在我的课题研究期间对我的热情帮 助。感谢公司李金岭老师、李成钢老师等在工作过程中的指导和帮助。感谢公司 的同事，陈婿、段征、杨帆、丁钊等，感谢你们在我工作遇到困难时给予我的各 种帮助和鼓励。

最后，感谢实验室的所有成员，三年中我们一起学习，在工作和学习上遇到 困难的时候能够一起面对，一起思考解决方法。感谢我的舍友们和朋友们，愿我 们的友谊长青。感谢我挚爱的父母，感谢你们长期以来的关心和鼓励，你们是我 永恒的后盾，谢谢你们。真诚地感谢所有在我工作和生活中给予帮助的人们！

杨本栋