爬虫的一些知识点

目录

[1. 网络爬虫 1](#_Toc7458)

[2. 产生背景 垂直领域搜索引擎 2](#_Toc14086)

[3. 1 聚焦爬虫工作原理以及关键技术概述 3](#_Toc19607)

[4. 涉及技术 3](#_Toc21725)

[4.1. 下载网页 一般是通过net api 3](#_Toc18703)

[4.2. 分析网页（html分析，接口可能有json 3](#_Toc19867)

[5. 分类 3](#_Toc17405)

[5.1. 通用网络爬虫（General Purpose Web Crawler）、聚焦网络爬虫（Focused Web Crawler）、增量式网络爬虫（Incremental Web Crawler）、深层网络爬虫（Deep Web Crawler）。 4](#_Toc7794)

[5.2. 聚焦网络爬虫 4](#_Toc14640)

[5.3. Deep Web 爬虫 4](#_Toc467)

[6. 网页分析算法 5](#_Toc22469)

[6.1. 网页分析算法可以归纳为基于网络拓扑、基于网页内容和基于用户访问行为三种类型。 5](#_Toc5899)

[7. 言归正传，java实现网络爬虫一般有五种方法 7](#_Toc25561)

[7.1. 2.基于HttpURLConnection类编写爬虫：java se的net包的核心类，主要用于http的相关操作。简单 7](#_Toc13246)

[7.2. 3.基于apache的HttpClient包编写爬虫：由net包拓展而来，专为java网络通信编程而服务。常用 7](#_Toc24207)

[7.3. 5.基于Selenium或者是WebDriver之类的有头（有界面）浏览器。。适合于复杂界面 8](#_Toc5713)

[8. 核心代码范例 8](#_Toc28507)

[8.1. 下载网页 8](#_Toc10366)

# 网络爬虫

（又被称为网页[蜘蛛](https://baike.baidu.com/item/%E8%9C%98%E8%9B%9B/8135707" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)，网络机器人，在[FOAF](https://baike.baidu.com/item/FOAF" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)社区中间，更经常的称为网页追逐者），是一种按照一定的规则，自动地抓取[万维网](https://baike.baidu.com/item/%E4%B8%87%E7%BB%B4%E7%BD%91/215515" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)信息的程序或者脚

爬虫目的：：为了获取数据，就是通过url接口 web gui接口来获取数据

操作web自动化（自动化注册，批量注册机，发帖机，点赞器）

自动化测试

# 产生背景 垂直领域搜索引擎

[编辑](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/javascript:;)

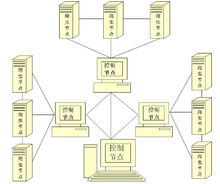
随着[网络](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)的迅速发展，[万维网](https://baike.baidu.com/item/%E4%B8%87%E7%BB%B4%E7%BD%91" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)成为大量信息的载体，如何有效地提取并利用这些信息成为一个巨大的挑战。[搜索引擎](https://baike.baidu.com/item/%E6%90%9C%E7%B4%A2%E5%BC%95%E6%93%8E" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)(Search Engine)，例如传统的[通用搜索引擎](https://baike.baidu.com/item/%E9%80%9A%E7%94%A8%E6%90%9C%E7%B4%A2%E5%BC%95%E6%93%8E" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)AltaVista，Yahoo!和Google等，作为一个辅助人们检索信息的工具成为用户访问万维网的入口和指南。但是，这些通用性搜索引擎也存在着一定的局限性，如：

(1)不同领域、不同背景的用户往往具有不同的检索目的和需求，通用搜索引擎所返回的结果包含大量用户不关心的网页。

(2)通用搜索引擎的目标是尽可能大的网络覆盖率，有限的搜索引擎服务器资源与无限的[网络数据](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E6%95%B0%E6%8D%AE" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)资源之间的矛盾将进一步加深。

(3)[万维网](https://baike.baidu.com/item/%E4%B8%87%E7%BB%B4%E7%BD%91" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)数据形式的丰富和[网络技术](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E6%8A%80%E6%9C%AF" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)的不断发展，图片、数据库、音频、视频多媒体等不同数据大量出现，[通用搜索引擎](https://baike.baidu.com/item/%E9%80%9A%E7%94%A8%E6%90%9C%E7%B4%A2%E5%BC%95%E6%93%8E" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)往往对这些信息含量密集且具有一定结构的数据无能为力，不能很好地发现和获取。

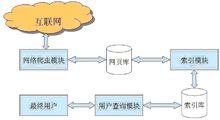
(4)通用搜索引擎大多提供基于[关键字](https://baike.baidu.com/item/%E5%85%B3%E9%94%AE%E5%AD%97" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)的检索，难以支持根据语义信息提出的查询。

[](https://baike.baidu.com/pic/ç½ç»ç¬è«/5162711/0/d833c895d143ad4bef92dd7b82025aafa50f06c8?fr=lemma%26ct=single)网络爬虫

为了解决上述问题，定向抓取相关网页资源的聚焦爬虫应运而生。聚焦爬虫是一个自动下载网页的程序，它根据既定的抓取目标，有选择的访问[万维网](https://baike.baidu.com/item/%E4%B8%87%E7%BB%B4%E7%BD%91" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)上的网页与相关的链接，获取所需要的信息。与通用爬虫(general purpose web crawler)不同，聚焦爬虫并不追求大的覆盖，而将目标定为抓取与某一特定主题内容相关的网页，为面向主题的用户查询准备数据资源。

# 1 聚焦爬虫工作原理以及关键技术概述

网络爬虫是一个自动提取网页的程序，它为[搜索引擎](https://baike.baidu.com/item/%E6%90%9C%E7%B4%A2%E5%BC%95%E6%93%8E" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)从万维网上下载网页，是搜索引擎的重要组成。传统爬虫从一个或若干初始网页的URL开始，获得初始网页上的URL，在抓取网页的过程中，不断从当前页面上抽取新的URL放入队列,直到满足系统的一定停止条件。聚焦爬虫的工作流程较为复杂，需要根据一定的网页分析算法过滤与主题无关的链接，保留有用的链接并将其放入等待抓取的URL队列。然后，它将根据一定的搜索策略从队列中选择下一步要抓取的网页URL，并重复上述过程，直到达到系统的某一条件时停止。另外，所有被爬虫抓取的网页将会被系统存贮，进行一定的分析、过滤，并建立索引，以便之后的查询和检索；对于聚焦爬虫来说，这一过程所得到的分析结果还可能对以后的抓取过程给出反馈和指导。

[](https://baike.baidu.com/pic/ç½ç»ç¬è«/5162711/0/500fd9f9d72a605987497aa12834349b033bba34?fr=lemma%26ct=single)

相对于通用网络爬虫，聚焦爬虫还需要解决三个主要问题：

(1) 对抓取目标的描述或定义；

(2) 对网页或数据的分析与过滤；

(3) 对URL的[搜索策略](https://baike.baidu.com/item/%E6%90%9C%E7%B4%A2%E7%AD%96%E7%95%A5" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)。

# 涉及技术

## 下载网页 一般是通过net api

## 分析网页（html分析，接口可能有json

# 分类

[编辑](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/javascript:;)

网络爬虫按照系统结构和实现技术，大致可以分为以下几种类型：

## 通用网络爬虫（General Purpose Web Crawler）、聚焦网络爬虫（Focused Web Crawler）、增量式网络爬虫（Incremental Web Crawler）、深层网络爬虫（Deep Web Crawler）。

实际的网络爬虫系统通常是几种爬虫技术相结合实现的 [1]  。

## 聚焦网络爬虫

聚焦网络爬虫（Focused Crawler），又称主题网络爬虫（Topical Crawler），是指选择性地爬行那些与预先定义好的主题相关页面的网络爬虫[8]。 和通用网络爬虫相比，聚焦爬虫只需要爬行与主题相关的页面，极大地节省了硬件和网络资源，保存的页面也由于数量少而更新快，还可以很好地满足一些特定人群对特定领域信息的需求 [1]  。

聚焦网络爬虫和通用网络爬虫相比，增加了链接评价模块以及内容评价模块。聚焦爬虫爬行策略实现的关键是评价页面内容和链接的重要性，不同的方法计算出的重要性不同，由此导致链接的访问顺序也不同 [1]  。

## Deep Web 爬虫

Web 页面按存在方式可以分为表层网页（Surface Web）和深层网页（Deep Web，也称 Invisible Web Pages 或 Hidden Web）。 表层网页是指传统搜索引擎可以索引的页面，以超链接可以到达的静态网页为主构成的 Web 页面。Deep Web 是那些大部分内容不能通过静态链接获取的、隐藏在搜索表单后的，只有用户提交一些关键词才能获得的 Web 页面。例如那些用户注册后内容才可见的网页就属于 Deep Web。 2000 年 Bright Planet 指出：Deep Web 中可访问信息容量是 Surface Web 的几百倍，是互联网上最大、发展最快的新型信息资源 [1]  。

Deep Web 爬虫体系结构包含六个基本功能模块 （爬行控制器、解析器、表单分析器、表单处理器、响应分析器、LVS 控制器）和两个爬虫内部数据结构（URL 列表、LVS 表）。 其中 LVS（Label Value Set）表示标签/数值集合，用来表示填充表单的数据源 [1]  。

Deep Web 爬虫爬行过程中最重要部分就是表单填写，包含两种类型：

1) 基于领域知识的表单填写：此方法一般会维持一个本体库，通过语义分析来选取合适的关键词填写表单。 Yiyao Lu[25]等人提出一种获取 Form 表单信息的多注解方法，将数据表单按语义分配到各个组中 ，对每组从多方面注解，结合各种注解结果来预测一个最终的注解标签；郑冬冬等人利用一个预定义的领域本体知识库来识别 Deep Web 页面内容， 同时利用一些来自 Web 站点导航模式来识别自动填写表单时所需进行的路径导航 [1]  。

2) 基于网页结构分析的表单填写： 此方法一般无领域知识或仅有有限的领域知识，将网页表单表示成 DOM 树，从中提取表单各字段值。 Desouky 等人提出一种 LEHW 方法，该方法将 HTML 网页表示为DOM 树形式，将表单区分为单属性表单和多属性表单，分别进行处理；孙彬等人提出一种基于 XQuery 的搜索系统，它能够模拟表单和特殊页面标记切换，把网页关键字切换信息描述为三元组单元，按照一定规则排除无效表单，将 Web 文档构造成 DOM 树，利用 XQuery 将文字属性映射到表单字段[1]  。

Raghavan 等人提出的 HIWE 系统中，爬行管理器负责管理整个爬行过程，分析下载的页面，将包含表单的页面提交表单处理器处理，表单处理器先从页面中提取表单，从预先准备好的数据集中选择数据自动填充并提交表单，由爬行控制器下载相应的结果页面 [1]  。

# 网页分析算法

[编辑](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/javascript:;)

## 网页分析算法可以归纳为基于[网络拓扑](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E6%8B%93%E6%89%91" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)、基于网页内容和基于用户访问行为三种类型。

**拓扑分析算法**

基于网页之间的链接，通过已知的网页或数据，来对与其有直接或间接链接关系的对象（可以是网页或网站等）作出评价的算法。又分为网页粒度、网站粒度和网页块粒度这三种。

1 网页(Webpage)粒度的分析算法

PageRank和[HITS算法](https://baike.baidu.com/item/HITS%E7%AE%97%E6%B3%95" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)是最常见的链接分析算法，两者都是通过对网页间链接度的[递归](https://baike.baidu.com/item/%E9%80%92%E5%BD%92" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)和规范化计算，得到每个网页的重要度评价。PageRank算法虽然考虑了用户访问行为的随机性和Sink网页的存在，但忽略了绝大多数用户访问时带有目的性，即网页和链接与查询主题的相关性。针对这个问题，HITS算法提出了两个关键的概念：权威型网页（authority）和中心型网页（hub）。

基于链接的抓取的问题是相关页面主题团之间的隧道现象，即很多在抓取路径上偏离主题的网页也指向目标网页，局部评价策略中断了在当前路径上的抓取行为。文献[21]提出了一种基于反向链接（BackLink）的分层式上下文模型（Context Model），用于描述指向目标网页一定物理跳数半径内的网页拓扑图的中心Layer0为目标网页，将网页依据指向目标网页的物理跳数进行层次划分，从外层网页指向内层网页的链接称为反向链接。

2 网站粒度的分析算法

网站粒度的资源发现和管理策略也比网页粒度的更简单有效。网站粒度的爬虫抓取的关键之处在于站点的划分和站点等级(SiteRank)的计算。SiteRank的计算方法与PageRank类似，但是需要对网站之间的链接作一定程度抽象，并在一定的模型下计算链接的权重。

网站划分情况分为按域名划分和按IP地址划分两种。文献[18]讨论了在分布式情况下，通过对同一个域名下不同[主机](https://baike.baidu.com/item/%E4%B8%BB%E6%9C%BA" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)、服务器的IP地址进行站点划分，构造站点图，利用类似PageRank的方法评价SiteRank。同时，根据不同文件在各个站点上的分布情况，构造文档图，结合SiteRank分布式计算得到DocRank。文献[18]证明，利用分布式的SiteRank计算，不仅大大降低了单机站点的算法代价，而且克服了单独站点对整个网络覆盖率有限的缺点。附带的一个优点是，常见PageRank 造假难以对SiteRank进行欺骗。

3 网页块粒度的分析算法

在一个页面中，往往含有多个指向其他页面的链接，这些链接中只有一部分是指向主题相关网页的，或根据网页的链接[锚文本](https://baike.baidu.com/item/%E9%94%9A%E6%96%87%E6%9C%AC" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)表明其具有较高重要性。但是，在PageRank和HITS算法中，没有对这些链接作区分，因此常常给网页分析带来广告等噪声链接的干扰。在网页块级别(Block?level)进行链接分析的算法的基本思想是通过VIPS网页分割算法将网页分为不同的网页块(page block)，然后对这些网页块建立page?to?block和block?to?page的链接[矩阵](https://baike.baidu.com/item/%E7%9F%A9%E9%98%B5" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)，?分别记为Z和X。于是，在page?to?page图上的网页块级别的PageRank为?W?p=X×Z；?在block?to?block图上的BlockRank为?W?b=Z×X。已经有人实现了块级别的PageRank和HITS算法，并通过实验证明，效率和准确率都比传统的对应算法要好。

**网页内容分析算法**

基于网页内容的分析算法指的是利用网页内容（文本、数据等资源）特征进行的网页评价。网页的内容从原来的以[超文本](https://baike.baidu.com/item/%E8%B6%85%E6%96%87%E6%9C%AC" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)为主，发展到后来[动态页面](https://baike.baidu.com/item/%E5%8A%A8%E6%80%81%E9%A1%B5%E9%9D%A2" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)（或称为Hidden Web）数据为主，后者的数据量约为直接可见页面数据（PIW，Publicly Indexable Web）的400~500倍。另一方面，多媒体数据、Web Service等各种网络资源形式也日益丰富。因此，基于网页内容的分析算法也从原来的较为单纯的[文本检索](https://baike.baidu.com/item/%E6%96%87%E6%9C%AC%E6%A3%80%E7%B4%A2" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)方法，发展为涵盖网页[数据抽取](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%8A%BD%E5%8F%96" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)、机器学习、[数据挖掘](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%8C%96%E6%8E%98" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)、语义理解等多种方法的综合应用。本节根据网页数据形式的不同，将基于网页内容的分析算法，归纳以下三类：第一种针对以文本和[超链接](https://baike.baidu.com/item/%E8%B6%85%E9%93%BE%E6%8E%A5" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)为主的无结构或结构很简单的网页；第二种针对从结构化的[数据源](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%BA%90" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)（如RDBMS）动态生成的页面，其数据不能直接批量访问；第三种针对的数据界于第一和第二类数据之间，具有较好的结构，显示遵循一定模式或风格，且可以直接访问。

基于文本的网页分析算法

1) 纯文本分类与聚类算法

很大程度上借用了[文本检索](https://baike.baidu.com/item/%E6%96%87%E6%9C%AC%E6%A3%80%E7%B4%A2" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)的技术。[文本分析](https://baike.baidu.com/item/%E6%96%87%E6%9C%AC%E5%88%86%E6%9E%90" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)算法可以快速有效的对网页进行分类和聚类，但是由于忽略了网页间和网页内部的结构信息，很少单独使用。

2) 超文本分类和聚类算法

根据网页链接网页的相关类型对网页进行分类，依靠相关联的网页推测该网页的类型。

# 言归正传，java实现网络爬虫一般有五种方法

（据我所知，要是有其他方法的同学欢迎分享）

1.基于socket通信编写爬虫：最底层的方式，同时也是执行最高效的，不过开发效率最低。

## 2.基于HttpURLConnection类编写爬虫：java se的net包的核心类，主要用于http的相关操作。简单

## 3.基于apache的HttpClient包编写爬虫：由net包拓展而来，专为java网络通信编程而服务。常用

4.基于phantomjs之类的无头（无界面）浏览器：

    (1)它是浏览器的核心，并非浏览器。换言之，它是没有UI的浏览器。

    (2)它提供的js api，故它可以方便直接的被各种程序语言调用。换言之，似乎是js写的。

## 5.基于Selenium或者是WebDriver之类的有头（有界面）浏览器。。适合于复杂界面

# 核心代码范例

## 下载网页

**import** org.apache.commons.io.IOUtils;

//爬虫核心 下载url网页使用jdk net api

**public** **class** demo {

**public** **static** **void** main(String[] args) **throws** IOException {

String urlInfo="http://www.baidu.com";

//读取目的网页URL地址，获取网页源码

URL url = **new** URL(urlInfo);

//打开http 网络连接

HttpURLConnection httpUrl = (HttpURLConnection)url.openConnection();

//获取input流

InputStream is = httpUrl.getInputStream();

//保存http的输出流到文件，持久化

IOUtils.*copy*(is, **new** FileOutputStream(**new** File("d:\\baidu.html")));

}