Beijing Forest Studio

北京理工大学信息系统及安全对抗实验中心

网络爬虫技术简介

门元昊　硕士研究生

2019年03月31日

内容提要

- 背景介绍
- 基本原理
- 关键技术
- 实际应用
- 拓展延伸
- 参考文献

背景介绍

- 网络爬虫的概念
  - 狭义上，指遵循标准的http协议，使用超链接和Web文档检索方法遍历万维网的软件程序。
  - 广义上，指能遵循http协议，检索Web文档的软件。
  - 简言之，网络爬虫是一种按照一定规则，自动的抓取网络上信息的程序或者脚本。
  - 又被称为网络蜘蛛或者网络机器人。

背景介绍

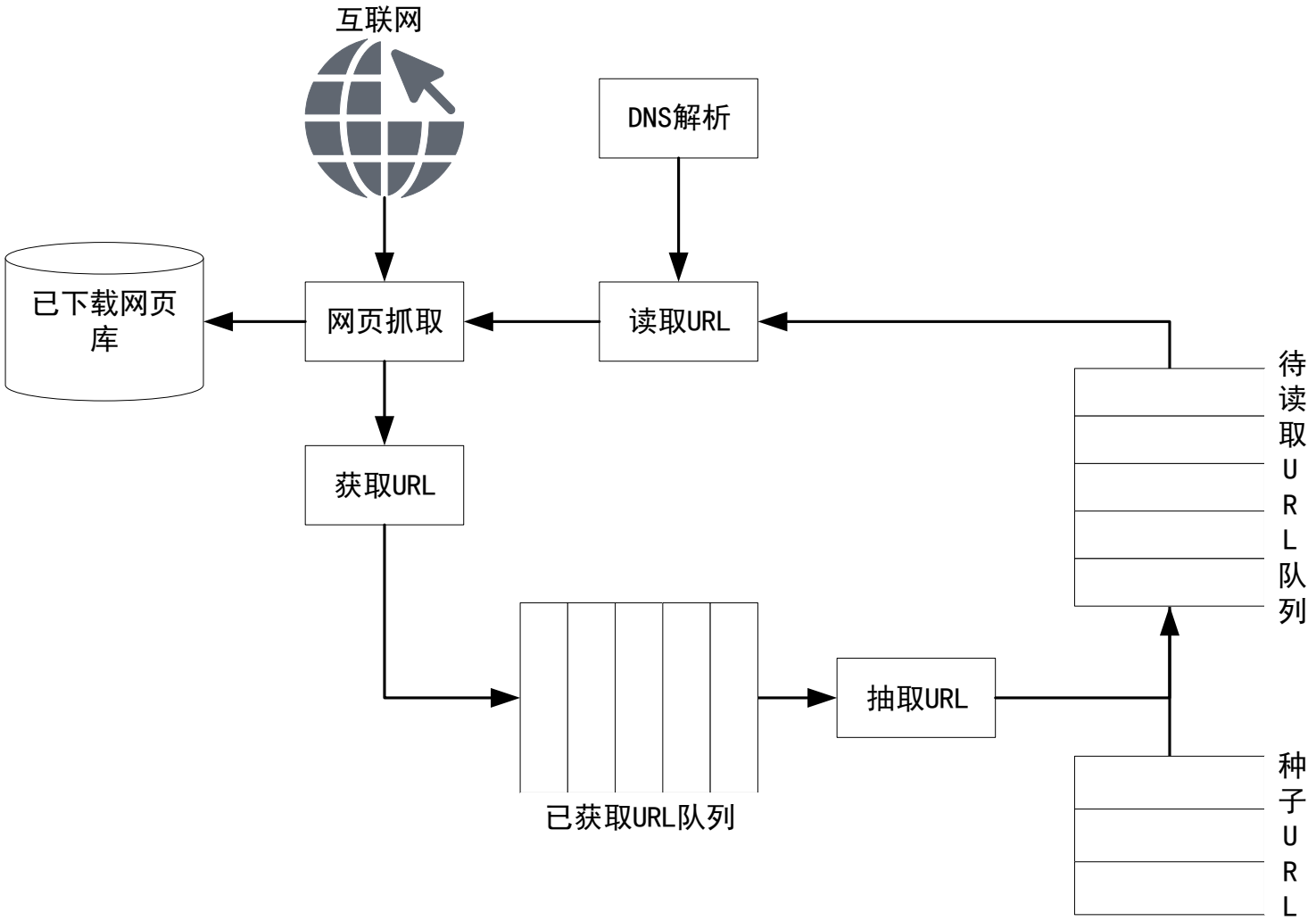
- 网络爬虫的常见用途
  - 宏观层面
    - 在互联网上寻找收集信息
    - 搜索引擎通过网络爬虫搜集网页的信息
    - 通过爬虫获取数据分析所需的大量数据
  - 个人层面
    - 爬学院、教务处通知信息
    - 爬淘宝、京东商品信息
    - 爬微博好友动态
    - ……

背景介绍

- 网络爬虫的分类
  - 通用爬虫
    - 从一个或若干初始网页的URL开始，先获得初始网页上的URL，在抓取过程中不断的从当前页面上获取新的URL放入队列，直到满足预设的停止条件。
  - 聚焦爬虫
    - 在通用爬虫的基础上进行改进。
    - 通过一定的网页分析算法过滤与主题无关的链接，将有用的链接放入等待抓取的URL队列中。
    - 根据一定的搜索策略从队列中选择下一步要抓取的URL进行抓取。

基本原理

- 网络爬虫结构框架图



关键技术

- 待抓取目标的定义与描述
- URL搜索策略
- 网页分析及信息提取

关键技术

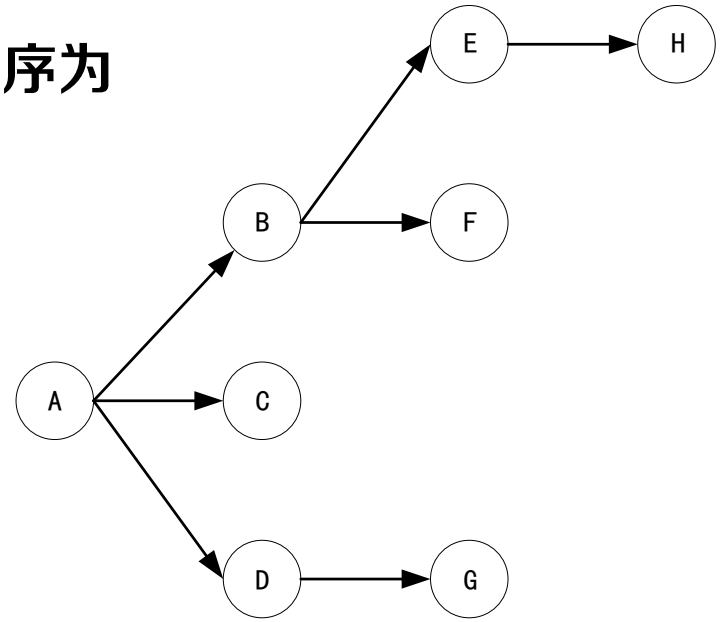
- 待抓取目标的定义与描述
  - 基于目标网页特征的网页级信息
    - 通常被通用搜索引擎采用
    - 用于抓取、存储并索引网站或网页，可能后续会提取需要的结构化信息
    - 便于维护但是灵活性较低
  - 基于目标数据模式的结构化数据
    - 常用于数据分析等方面
    - 用于爬取所需的数据
    - 灵活性高但是修改维护成本较高

关键技术

- 网页搜索策略
  - IP遍历搜索策略
  - 广度优先搜索策略
  - 深度优先搜索策略
  - 最佳优先搜索策略

关键技术

- 网页搜索策略
  - 广度优先搜索策略
    - 在抓取过程中，在完成当前层次的搜索后，才进行下一层次的搜索。
    - 按照本策略，右图的遍历顺序为
    - A->B, C, D, E, F, G, H

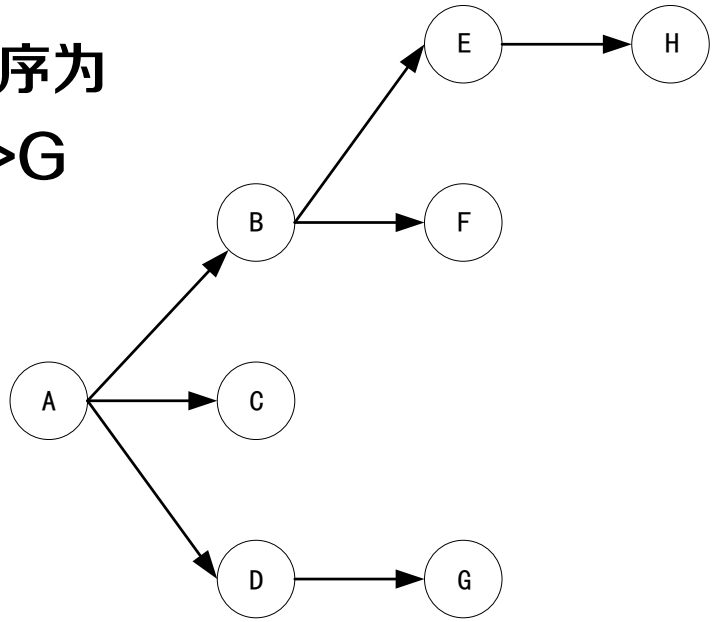


关键技术

- 网页搜索策略
  - 广度优先搜索策略
    - 优点
      - 设计实现相对简单，并可以快速覆盖尽可能多的网页。
    - 缺点
      - 随着抓取网页的增加，大量无关的网页将被下载并处理。

关键技术

- 网页搜索策略
  - 深度优先搜索策略
    - 先沿着Web文件上的超链接一个个链接的深入，然后返回至原Web文件，继续选择下一个链接深入。
    - 按照本策略，右图的遍历顺序为
    - A->B->E->H, F, C, D->G



关键技术

- 网页搜索策略
  - 深度优先搜索策略
    - 优点
      - 逻辑容易设计，便于实现
    - 缺点
      - 容易使爬虫出现陷入(trapping)问题
        - » 由于Web结构过深，爬虫会一直向下爬取的问题。

关键技术

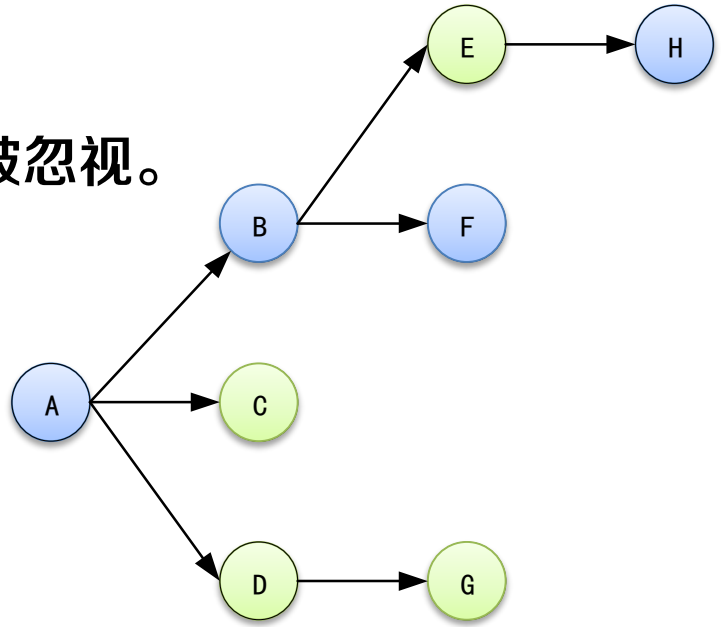
- 网页搜索策略
  - 最佳优先搜索策略
    - 按照一定的网页分析算法，先计算网页的相似度，然后根据预先设置的阈值对合格的URL进行抓取。
    - 优点
      - 经过调整的最佳优先搜索策略可以将无关网页数量降低30%~90%
    - 缺点
      - 是一种局部最优的最优算法
      - 容易忽略爬取路径上的很多相关网站

关键技术

- 网页搜索策略
  - 最佳优先搜索策略

    - A, B, F, H为想要爬取的高相似度网页。
    - C, D, E, G为低相似度网页。

    - 在实际爬取中，H网页通常会被忽视。



关键技术

- 网页分析及信息提取（聚焦爬虫）
  - 基于网络拓扑关系的分析算法
    - 基本概念
      - 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作出评价的算法。
    - 常见算法
      - 网页粒度分析算法
        - » 通过对网页间链接度的递归和规范化计算，得到每个网页的重要度评价。
      - 网站粒度分析算法
        - » 划分站点的层次和等级，而不再具体的计算站点下的各个网页的等级。

关键技术

- 网页分析及信息提取
  - 基于网页内容的分析算法
    - 基本概念
      - 利用网页内容（文本、数据等资源）特征进行的网页评价。
    - 常用算法
      - 基于文本的分析算法
        - » 纯文本
        - » 超文本
      - 结合数据挖掘、机器学习等技术的分析算法

实际应用

- 常用的爬虫框架
  - Scrapy
  - Pyspider
  - Requests + BeautifulSoup
  - Requests-html
  - Nutch
  - Webmagic
  - PhpSpider
  - open-source-search-engine
- Pyspider的简单程序演示

实际应用

- 常用的爬虫框架
  - Scrapy
    - 基于Python开发的一个快速、高层次的web抓取框架，用于抓取web站点并从页面中提取结构化的数据。
    - 优势
      - 定制化程度高
      - 可扩展性强，支持插件
      - 可移植性高
    - 劣势
      - 爬虫编写时间较长
      - 不支持分布式
      - 只保存抽取结果，默认不保存网页

实际应用

- 常用的爬虫框架
  - Pyspider
    - 由国人基于Python开发的一个网络爬虫框架，带有WebUI。
    - 优势
      - 有图形界面，操作简单明了
      - 简单易用，仅需编写爬取的规则
      - 对新手友好
    - 劣势
      - Windows下会出现安装依赖问题（可能）
      - 社区支持与Scrapy相比较弱
      - 定制化程度较低

实际应用

- 常用的爬虫框架
  - Requests + BeautifulSoup
  - Requests-html

    - 并非爬虫框架，而是Python的http请求和html解析库，需要自己编写抓取程序和解析程序。
    - 优势
      - 定制化程度最高，可以完全按照自己的意愿进行编写。
      - 开发速度快。
    - 劣势
      - 爬虫程序的功能和性能严格依赖于编写者的水平。

实际应用

- 常用的爬虫框架
  - Nutch
    - 一个基于Java的开源搜索框架。隶属于Apache基金会，最初服务于Lucene搜索引擎框架，主要用于通用数据的爬取。
    - 优势
      - 泛用性强，适用于各种规模数据的爬取
      - 支持分布式爬取，提供分布式调度和存储功能
      - 拥有插件框架，可以实现各类定制化爬取功能
      - 支持与Hadoop集群对接
    - 劣势
      - 插件编写较繁琐，定制化成本高
      - 默认不抓取动态网页

实际应用

- 常用的爬虫框架
  - WebMagic
    - 简单灵活的Java爬虫框架。
    - 优势
      - API简单，便于新人快速上手
      - 扩展性强
      - 支持多线程爬取
      - 支持分布式爬取
    - 劣势
      - Java框架...

实际应用

- 常用的爬虫框架（并不是）
  - PhpSpider
    - 基于Php开发的爬虫框架
  - open-source-search-engine
    - 基于C/C++开发的网络爬虫和搜索引擎

实际应用

- Pyspider的简单程序演示

拓展延伸

- 反爬虫技术
  - 屏蔽右键
    - 一种简单粗暴的方案，通过阻止用户在浏览器中使用右键菜单检查元素来防止用户对网站的分析。
  - IP限制
    - 通过封禁短时间内大量访问的IP，来阻止爬虫的抓取。

拓展延伸

- 反爬虫技术
  - User-Agent检查
    - User-Agent的概念
      - 用户代理，又称UA，是http请求中的一个特殊字符串，用以帮助服务器识别用户使用的操作系统及版本、CPU类型、浏览器及版本、浏览器渲染引擎、浏览器语言、浏览器插件等。
    - User-Agent白名单
      - 通常，网站都会建立 user-agent白名单，只有属于正常范围的user-agent才能够正常访问。

拓展延伸

- 反爬虫技术
  - JS方法
    - JS脚本
      - 例如使用JS代码随机生成一组数，要求浏览器通过JS的运算求和，然后返回服务器验证。
    - 验证码
      - 图片验证码
      - 人机验证
    - 数据异步加载
      - 使用Ajax异步加载数据，使得页面源代码中不存在数据。

拓展延伸

- 反反爬虫技术
  - IP代理池
    - 付费
    - 免费
  - User-Agent池
  - 简单的验证码识别技术

拓展延伸

- 反反爬虫技术
  - Selenium
    - 一个用于Web应用程序测试的工具，可以被用于网络爬虫。
    - 测试直接运行在浏览器中，就像真正的用户在操作。
    - 支持包括IE（7, 8, 9, 10, 11），Mozilla Firefox，Safari，Google Chrome，Opera等的浏览器。
    - 支持包括Python, Java, C#, Javascript(Nodejs), Php等多种语言。

拓展延伸

- 反反爬虫技术
  - Selenium
    - 优点
      - 完全模拟用户操作，可以屏蔽掉一大部分反爬虫方法
      - 支持多语言多平台，可移植性高
    - 缺点
      - 过于笨重，爬取速度较慢

参考文献

- [1]https://blog.csdn.net/gan18662672213/article/details/80967242
- [2]https://server.zzidc.com/fwqjs/2242.html
- [3]https://www.seleniumhq.org/docs/01_introducing_selenium.jsp

道德经

大成若缺，其用不弊。
大盈若冲，其用不穷。
大直若屈。大巧若拙。
大辩若讷。静胜躁，寒胜热。清静为天下正。

谢谢！