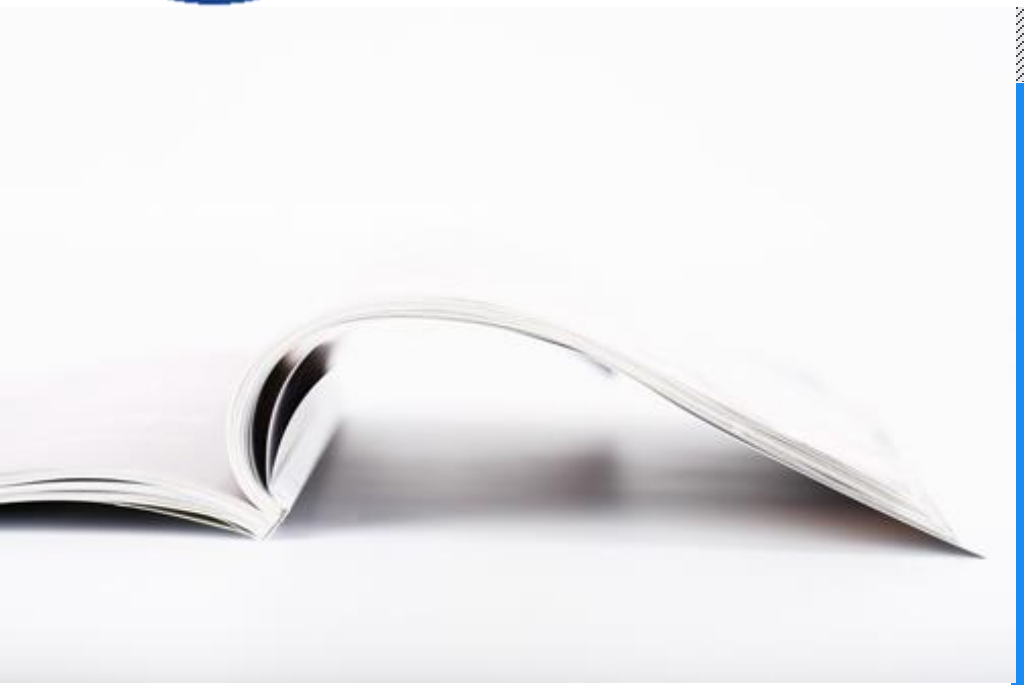


Web信息处理与应用

第二节

爬虫基础

徐童

2023.9.11

上节回顾

- 1965-1989，两位先驱的成果

- Ted Nelson在1965年提出了超文本的概念。
  - HyperText，源自于“非连续性著述”（Non sequential writing）的理念，即分叉的、允许读者作出选择的、无限延伸扩展的非线性文本。
- 1989年，Tim Berners Lee（万维网之父）等人首次提出了一个World Wide Web协议，随后定义了URL、HTML、HTTP等的规范，使网络能够为大众所使用。

上节回顾

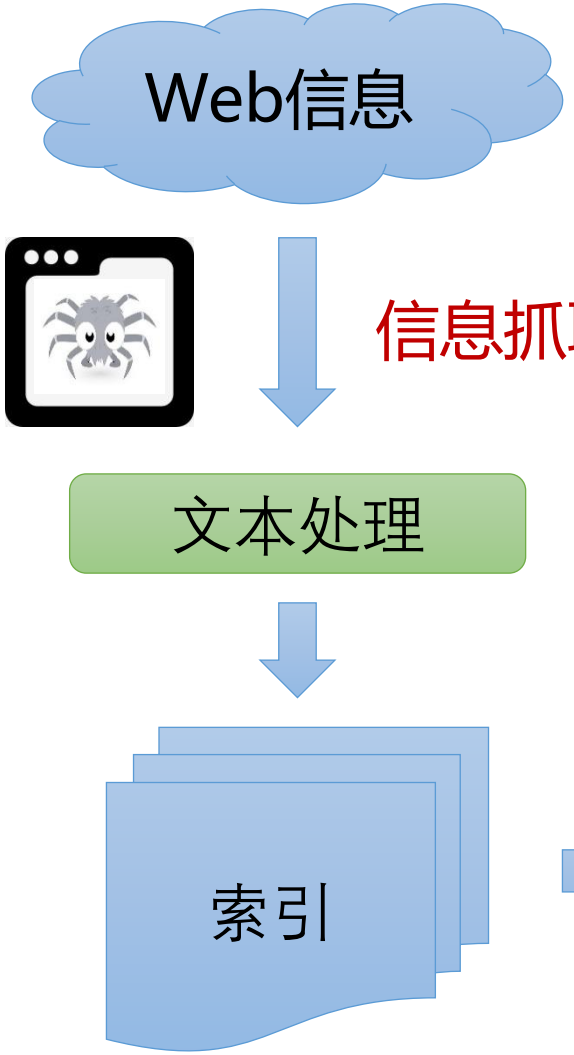
- 1993年，脱胎于爬虫的搜索引擎Wanderer

- Wanderer：最早的爬虫
  - 由MIT的学生Matthew Gray设计
  - 原意用于统计互联网上服务器的数量，而非为搜索引擎所设计
- Wandex：最早的网页索引计划
  - Wanderer后来发展为可以捕获网址，而为这些网址建立索引的计划就是Wandex

Wandex

本节主题

- 本课程所要解决的问题



第一个问题：

Web信息如何获取?

推荐课程

- **写在前面：本节课程说明**

- 本节课以爬虫基本概念和基础知识为主
  - 计算机课？语文课！（滑稽  ）
- 有想要了解更多关于爬虫实战内容的同学，可参考以下在线课程：
  - Python教程（第21节-第29节部分）
    - https://www.bilibili.com/video/BV1ws411i7Dr/
  - Python爬虫全套课程
    - https://www.bilibili.com/video/BV1Yh411o7Sz

推荐课程

- **TA倾情推荐**

- 如果觉得听课太累，下面还有更简洁的实战教程
  - 零基础入门版

    https://zhuanlan.zhihu.com/p/21479334
  - 常用工具
    - https://zhuanlan.zhihu.com/p/110448373
    - https://zhuanlan.zhihu.com/p/57678048

推荐课程

- **TA倾情推荐**

- 实战教程
  - 卷王版
    - 一个收集各种爬虫 （默认爬虫语言为 python）的集合（如Bilibili用户、京东商品等），一应俱全。不过有的失效或者不更新了，大佬们自由发挥。

    https://github.com/facert/awesome-spider

小卷怡情，大卷伤身，强卷灰飞烟灭，内卷需谨慎

推荐课程

- **TA倾情推荐**

- 进阶版：分布式爬虫
  - 含豆瓣热门、百度贴吧、百度翻译3个实例。建议循序渐进。

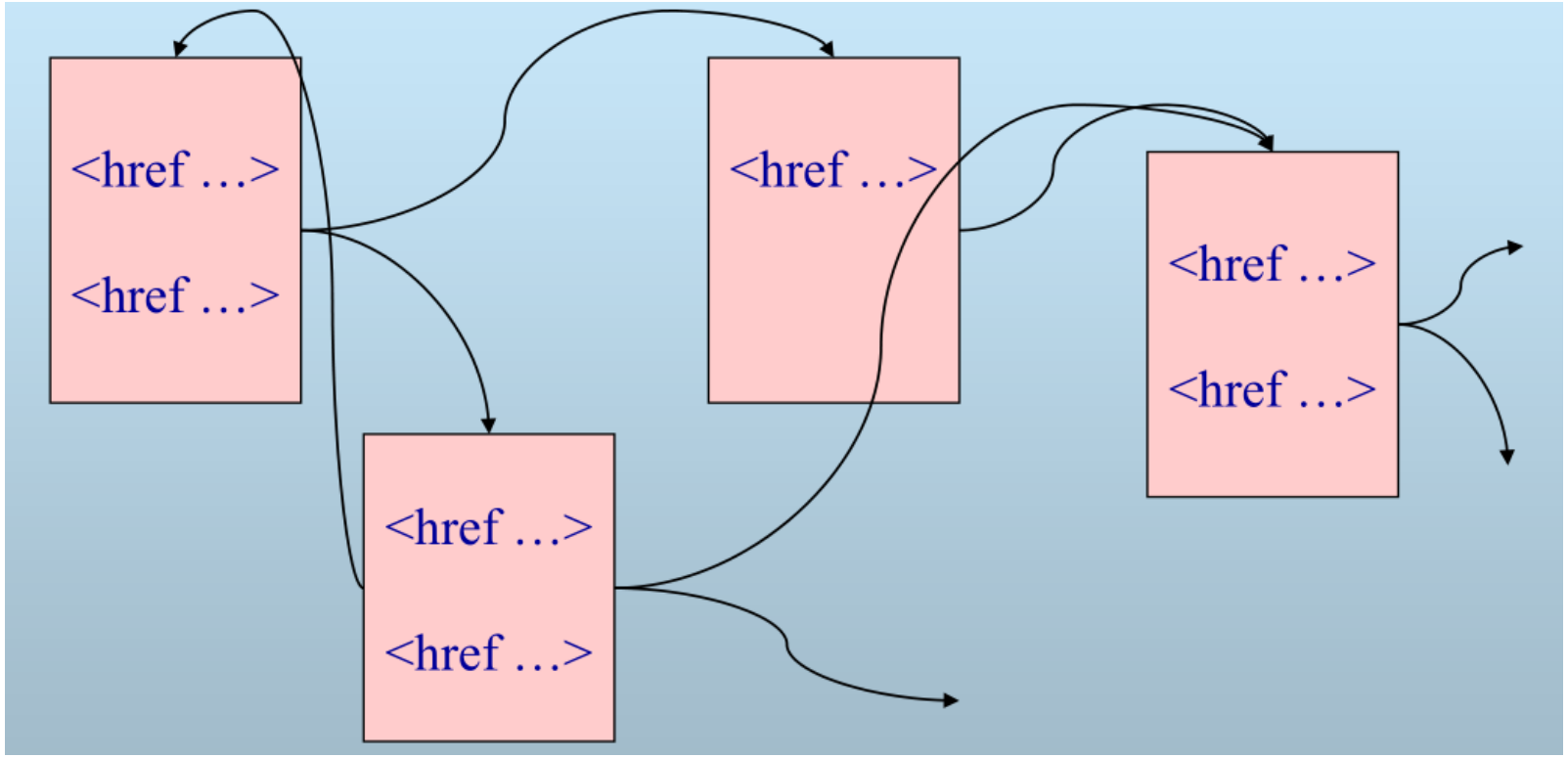
  https://github.com/Kr1s77/Python-crawler-tutorial-starts-from-zero
- 拓展（移动端数据如何获取？）
  - 微信小程序 Charles抓包
    https://blog.csdn.net/HeyShHeyou/article/details/90452656
  - 手机APP fiddler抓包

    https://zhuanlan.zhihu.com/p/34430703

本节目录

- **网络爬虫的定义与需求**
- 爬虫的基本要素
- 面向API的新爬虫任务
- 常见的爬虫算法
- 常见反爬虫机制与应对策略

- **Web网络的图模型**
- 以网页为节点、超链接（Hyper-Link）为有向边



- **爬虫的任务定义**

- 从一个种子站点集合（Seed sites）开始，从Web中寻找并且下载网页，获取排序需要的相关信息，并且剔除低质量的网页。
- 常见的爬虫类型
  - 通用网络爬虫：目标为全网Web信息，主要为门户站点搜索引擎和大型Web 服务提供商采集数据。
  - 聚焦网络爬虫：选择性爬取与预定主题相关的内容。
  - 增量式网络爬虫：对已下载内容进行增量式更新并只爬取更新内容。

- **爬虫的任务定义**

- 从一个种子站点集合（Seed sites）开始，从Web中寻找并且下载网页，获取排序需要的相关信息，并且剔除低质量的网页。
- 常见的爬虫类型
  - 深度网络爬虫：专门负责获取搜索引擎无法索引的、超链接不可达的或需提交表单后才可见的网络内容。

- **爬虫的基本用途**

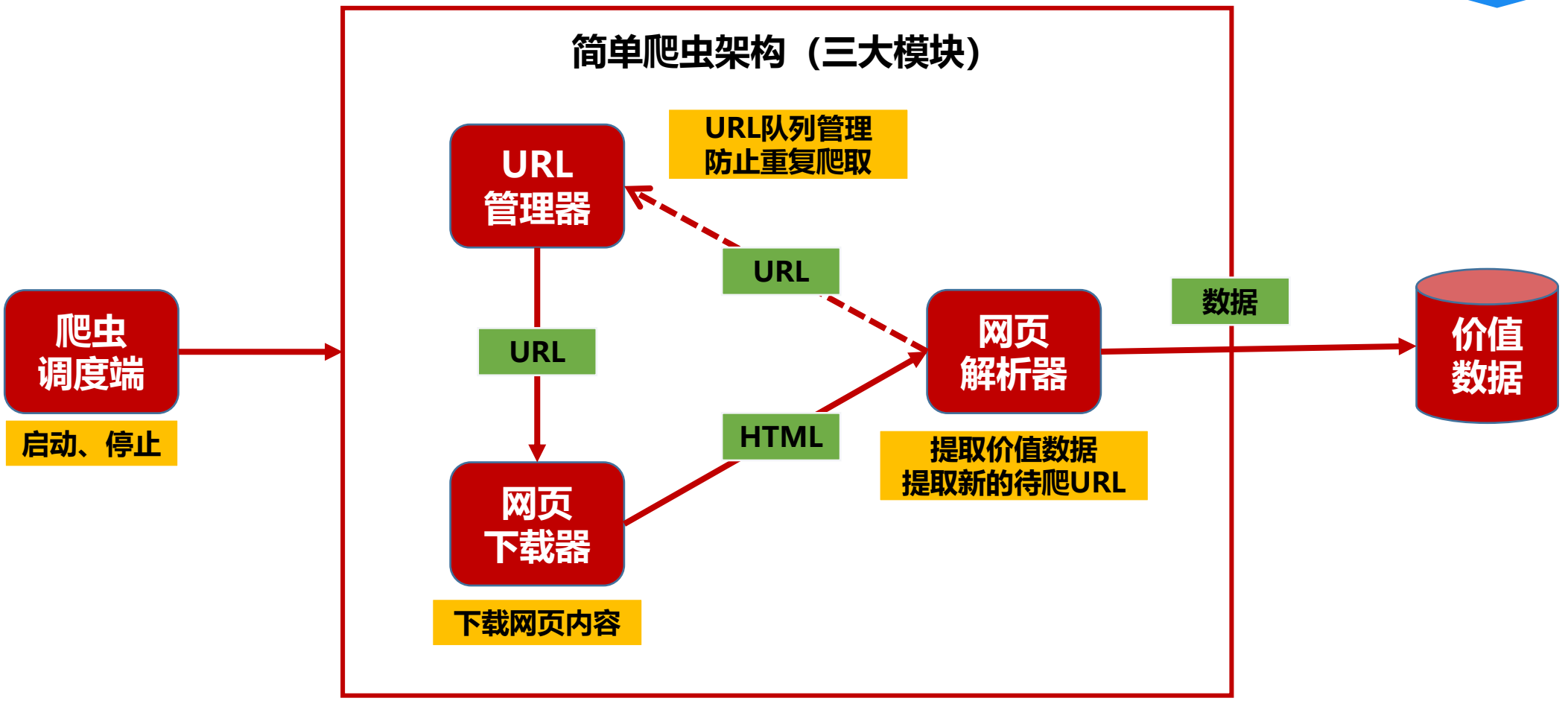
数据展示

引擎优化

特定应用

数据分析

- **爬虫的基本流程**



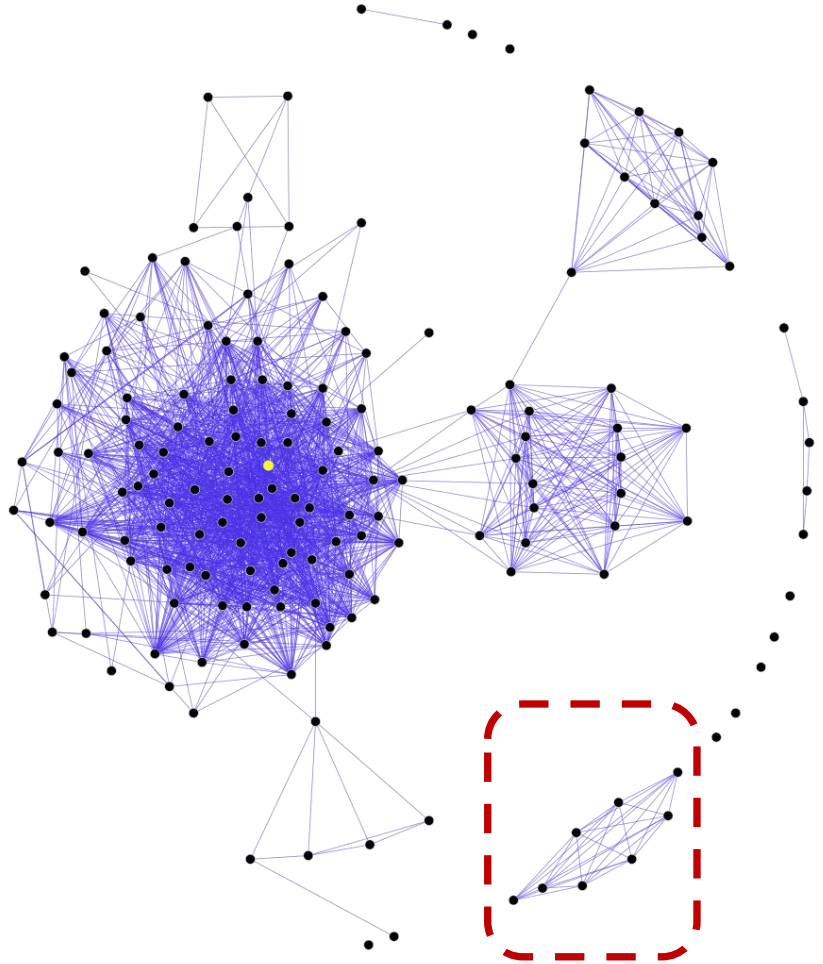
从爬取网页中获取更多URL，充实URL库，进而获取新的网页

- **网络爬虫的性能衡量**

- 数量覆盖率：“全”
  - 搜索引擎索引的网页，占目标区域中所有可能网页数量的百分比
- 质量覆盖率：“好”
  - 搜索引擎索引的网页中，“高质量”网页占目标区域中所有重要网页的百分比

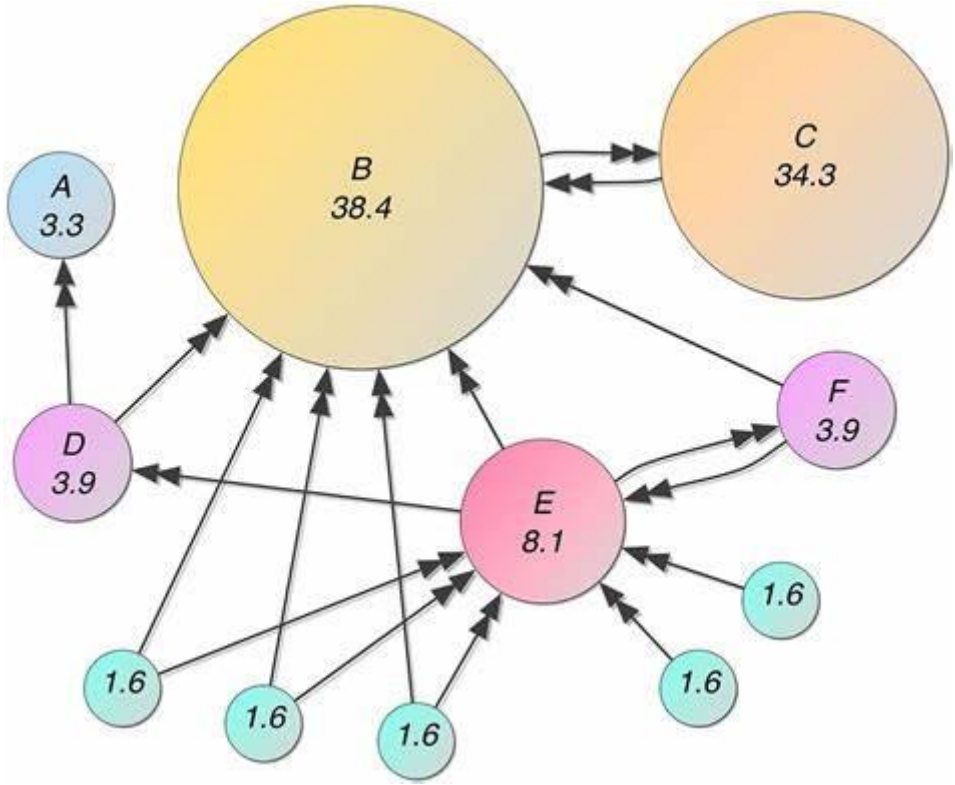
- **爬虫的数量覆盖率问题——“全”**

- 遍历完备性无法保证
  - 孤立节点的存在（如右图右半部分）
- 其他影响遍历的因素
  - 部分“偏远”节点难以遍历
  - 网络结构的动态演化
  - IP/Robots等条件的约束



- **爬虫的质量覆盖率问题——“好”**

- 如何度量网页的重要性?
  - 启发式方法：出入度…
  - 基于结构：PageRank/HITS…
  - 开放问题：信息密度衡量?
- 不仅重要，还要保证时效
  - “时新性”的要求



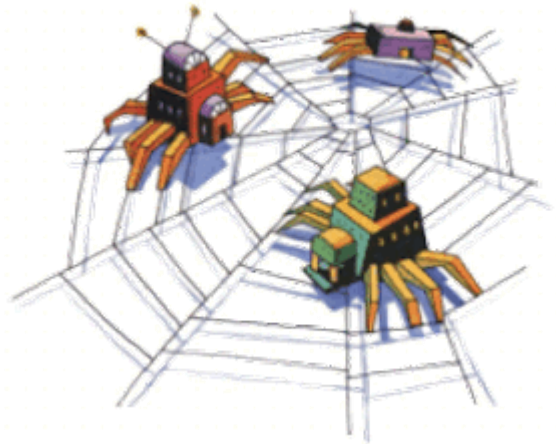
- **爬虫的主要需求**
- 速度（Speed）：突破网络瓶颈，最短时间内获取所需网页
- 可扩展性（Scalability）：基于多爬虫机制，有效打破单爬虫效率天花板
- 健壮性（Robustness）：有效应对服务器陷阱等潜在问题
- 时新性（Freshness）：保持内容的时效性，提升用户体验
- 友好性（Politeness）：遵循网络秩序，不影响网络服务正常运转
- 合法性（Legitimacy）：遵守相关法律，不侵犯用户隐私、商业机密等

- **爬虫需求：速度**

- 2012年的统计数据：谷歌每天需要爬取200亿个页面
  - 以每周1400亿个页面计算，约为2^{37}个页面
  - 平均每个页面64KB，约为2^{16}个bit
    - 实际情况远远不止64KB，大量非文本内容
  - 可知每秒流量>100Gb/s
- 网络瓶颈的存在对速度的限制
  - 以家用100Mbps网络为例，考虑以太网MTU为1500字节（上限了！），即使达到100%利用率，每秒也仅传输约8333个网页。

- **爬虫需求：可扩展性**

- 单一爬虫工作效率低下，难以突破效率天花板。
- 采用多爬虫机制已成为大型搜索引擎的工作常态。
- 多个爬虫带来新的挑战
  - 如何管理多个并发的连接?
  - 不同爬虫负责不同URL，如何进行划分?
  - 过多的硬件并行好处并不大
    - 抓取的性能瓶颈出现在通讯和硬盘读写



- **爬虫需求：健壮性**

- 如何有效应对爬取过程中所面临的各种风险？
  - 爬取网页时陷入回路怎么处理？
  - URL不规范如何解决？
  - 服务器陷阱如何应对？
  - 系统崩溃如何处理？

www.troutbums.com/Flyfactory/hatchline/hatchline/hatchline/flyfactory/flyfactory/flyfactory/flyfactory/flyfactory/flyfactory/flyfactory/flyfactory/flyfactory/hatchline

- **爬虫需求：时新性**

- T时刻的时新性：所抓取的网页内容与T时刻该网页的最新内容一致
- 网页年龄：距离网页最近一次更新的时间
  - 通过对网页更新行为的建模来预测年龄
- 保持爬取内容的时新性能够提升搜索效果
  - 但同时，过度重视时新性将严重增加搜索引擎的负担。
  - 部分网页频繁更新，增加抓取难度

- **爬虫需求：友好性**

- 不能显著影响被爬取的服务器性能
  - 大量DNS查询可能造成类似DOS的效果
- 有些服务器可能不希望被爬取
  - Robots exclusion
  - 反面教材：360搜索、头条搜索

- **爬虫需求：合法性**
- 法律中的规定
  - 我国：《互联网搜索引擎服务自律公约》，无针对爬虫的特定规定，但爬取的数据可能侵犯用户隐私权、违反《反不正当竞争法》等
  - 欧盟：GDPR
  - 加州：CCPA
  - ……

数据之争：网络爬虫涉及的法律问题 - 金杜律师事务所

- **爬虫需求：合法性**

- **Linkedin vs hiQ （2019）**

  hiQ 作为数据分析公司爬取 Linkedin 的数据，Linkedin 发出要求终止的信函之后限制了 hiQ 访问数据。最终最高法院判决 Linkedin 不得作出限制。

- **杭州魔蝎科技侵犯公民个人信息案（2021）**

  魔蝎科技将开发的前端插件嵌入网贷平台App中，利用各类爬虫技术，爬取用户本人账户内的通话记录、社保、公积金等各类数据，并提供给网贷平台用于判断用户的资信情况

Watch-outs for Legal and Ethical Web Scraping in 2022

- **爬虫需求：友好 & 合法**

一个合适的爬虫是怎样的

- 只爬取需要的数据
- 满足被爬站点的使用协议
  - 注意：Robots协议被认定为搜索引擎行业内公认的、应当被遵守的商业道德
- 不对被爬网站造成不良影响（如造成DDoS）
- 不收集可定位到个人的信息（数据脱敏/合并）
- 保护好被爬取的数据
- ……

Watch-outs for Legal and Ethical Web Scraping in 2022

- **最后，老师和助教有话说**

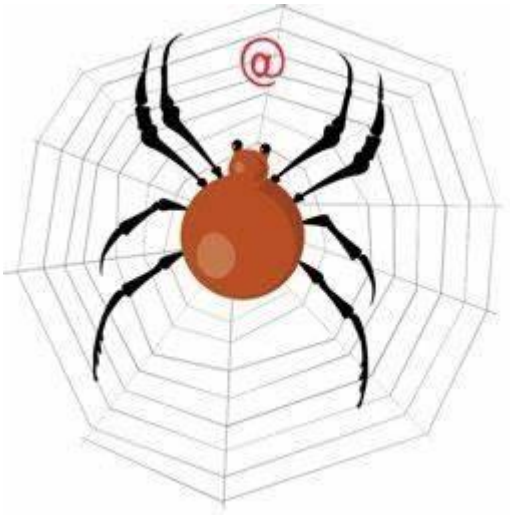
- 友情提示：爬虫本中立，数据应保护
  - 遵守协议，不要爬取隐私、涉密数据
  - 爬取方式应友好且合法，不影响网站正常运行
  - 如果不幸吃上大碗牢饭，不把为师说出去就行了

本节目录

- 网络爬虫的定义与需求
- **爬虫的基本要素**
- 面向API的新爬虫任务
- 常见的爬虫算法
- 常见反爬虫机制与应对策略

- **爬虫所涉及的协议和要素**

- HTML / HTTP
- DNS / URL
- Sitemap
- Robots Exclusion



HTML

- 如何表示与获取网页中的链接结构?

- **HTML文件示例**

```
▶ <div class="container">…</div>
▼ <nav class="main-menu">
  ▼ <div class="container" style="z-index: 10;">
      <a class="toggle"></a>
    ▼ <ul class="menu">
      ▼ <li>
          <a href="xxgk/xxjj.htm" title="学校概况">学校概况</a>
        ▼ <ul>
          ▼ <li>
              <a href="xxgk/xxjj.htm" title="学校简介">学校简介</a>
            </li>
          ▼ <li>
              <a href="xxgk/xrld.htm" title="现任领导">现任领导</a>
            </li>
          ▼ <li>
              <a href="xxgk/gljg.htm" title="管理机构">管理机构</a>
            </li>
          ▼ <li>
              <a href="xxgk/xydt.htm" title="校园地图">校园地图</a>
            </li>
          </ul>
        </li>
      ▼ <li>
          <a href="yxjs.htm" title="院系介绍">院系介绍</a>
        </li>
      ▼ <li>
          <a href target="_blank" title="师资队伍">师资队伍</a>
        ▼ <ul>
```

HTML的基本概念

- HyperText Markup Language，书写网页的“框架语言”
- 基本组成：“标记”（Tags）+“文本内容”（Text）
  - 例如：学校概况
- 标记的作用：
  - 说明网页元数据（如“标题”等）
  - 说明文本内容的布局和字体、字号等信息
  - 嵌入图片、视频、创建超链接等

- **HTML的示例框架**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
<head>
        <title> This is the title but often omitted </title>
</head>
<body>
        <img src = “url1” alt=“text”>
        other text
        <a href = “url2” title=“anchor text”> this is link text </a>
</body>
</html>
```

- **HTML中具有特别意义的文字**

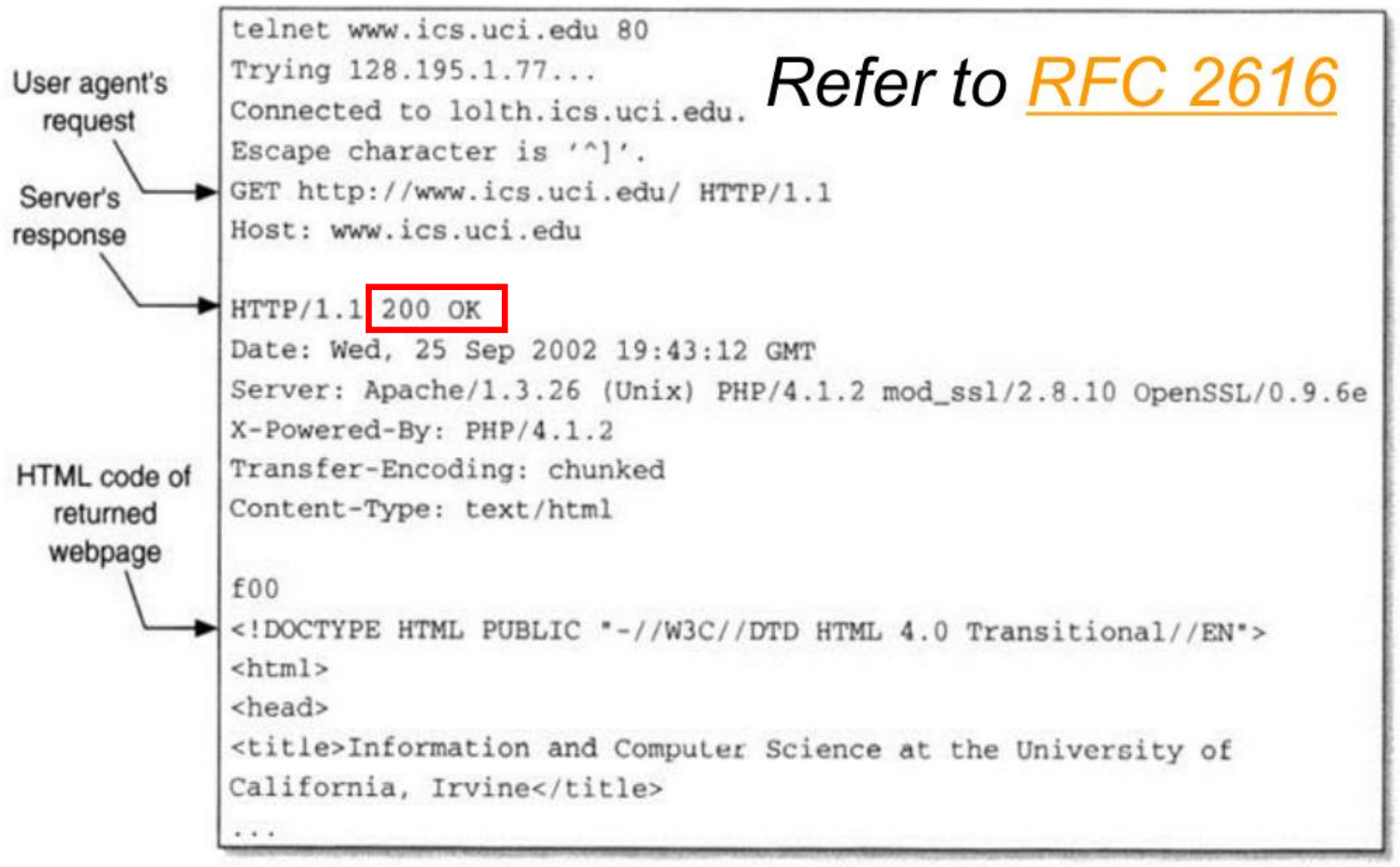
- <head><title> text </text></head>
- 是搜索服务显示的内容之一（URL，标题，摘要等）
- <img src=“url” alt=“text”>
- 图片描述，可以帮助我们做“从文字到图片”的查询
- link text
- 有助于理解目标网页内容及网页之间在内容上的关系

HTTP

HTTP的基本概念

- 超文本传输协议（HyperText Transport Protocol）
  - 工作在TCP 之上（请求/应答方式）
  - 容许在一个TCP 连接上发多个HTTP请求
- 工作步骤（从客户端看）
  - 通过域名服务器（DNS ）得到服务器主机的IP地址
  - 用TCP和服务器建立联系
    - 发送HTTP 请求（例如，GET或POST）
    - 接收HTTP应答头
    - 接收HTML网页内容

- **HTTP的应用实例**



Example of the use of the GET method in an HTTP 1.1 session.

HTTP

• HTTP的状态代码分类

分类	分类描述
1**	信息，服务器收到请求，需要请求者继续执行操作
2**	成功，操作被成功接收并处理
3**	重定向，需要进一步的操作以完成请求
4**	客户端错误，请求包含语法错误或无法完成请求
5**	服务器错误，服务器在处理请求的过程中发生了错误

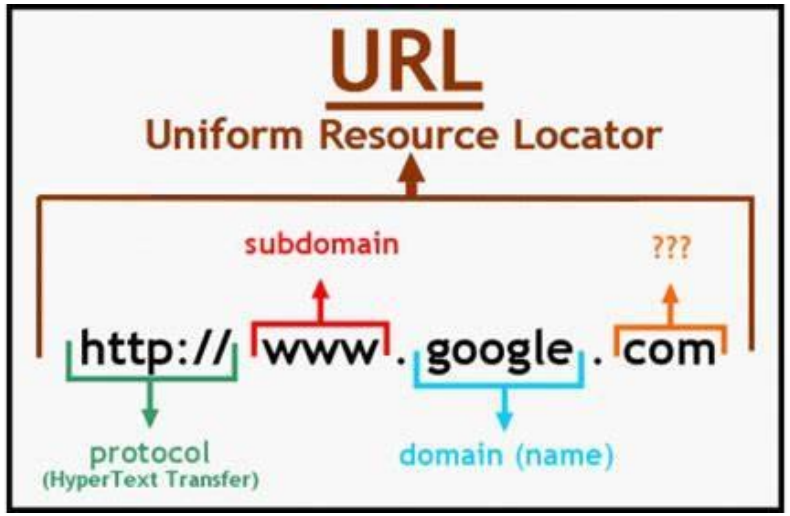
重点记住一些特殊的状态码，如：200（OK），403（Forbidden），404（Not Found）……

URL

- **URL的基本概念**

- 统一资源定位符（Universal Resource Locator）
  - 以某种统一的（标准化的）方式标识资源的简单字符串。

- URL一般由四部分组成：
  - 访问资源的模式/协议。
  - 存放资源的主机名。
  - 资源自身的名称，由路径表示。
  - 被访问的文件名或主页名。



- **URL实例**

- URL例子：http://cs.ustc.edu.cn/3058/list.htm

  - 这是一个可通过HTTP协议获得的文档。
  - 存放在名为cs.ustc.edu.cn的主机名。
  - 通过路径3058（对应学院新闻栏目）访问。

URL与IP地址的对应关系

- 一般而言，域名与IP地址处于一一对应的关系
- 实际情况下，域名与IP地址之间存在复杂的对应关系
  - 一对一：基本的对应关系
  - 一对多：可能由于虚拟主机导致，使得多个URL映射到同一IP
    - 例如，www.pku.edu.cn / www.gh.pku.edu.cn 等都映射到162.105.129.12上，但由于各个站点内容不同，所以被认为是不同的Web服务器
  - 多对一：可能由于DNS轮询导致，以应对商业站点的负载问题
    - 例如，多个182.61.200.x均与www.baidu.com 映射。

- **URL链接提取与规范化**

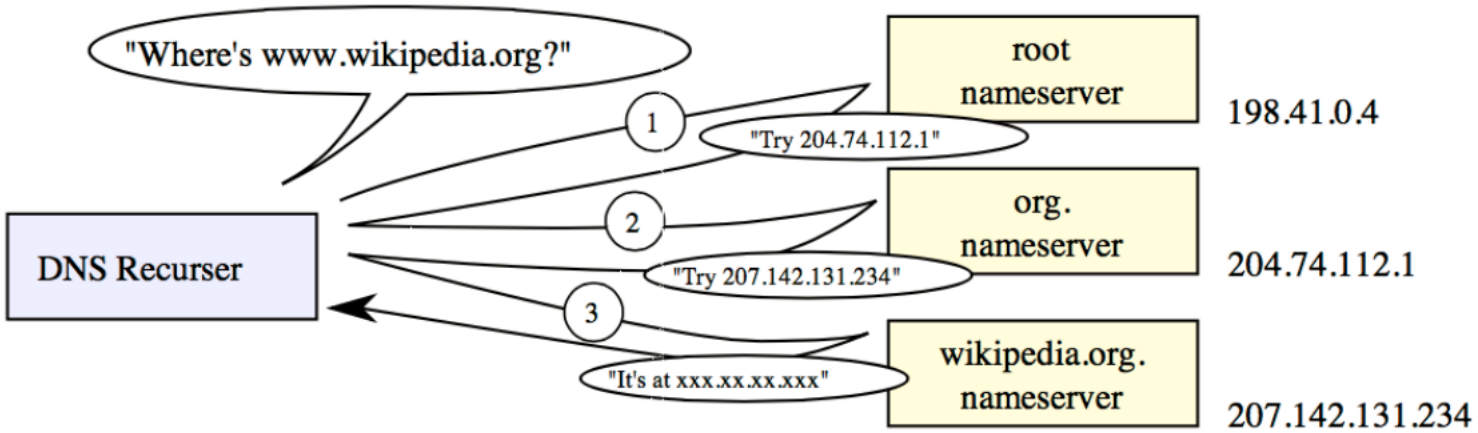
- 目标：得到网页中所含URL的标准型
  - URL的处理与过滤，避免多次获得不同URL指向的相同网页
  - 相对URL：基础URL被省略，需要进行补齐
- URL的不规范现象
  - 不同URL可能指向同一个网页
  - URL的格式存在错误或无用内容
    - 多余的文件名（如index.html），无用的查询变量或空的查询条件
  - 动态网页与动态参数
  - **短链接**

- **一些URL规范化的操作**

- URL的基本组成：协议:// 主机名[: 端口]/ 路径/[: 参数] [? 查询]#Fragment
  - protocol ://hostname[:port]/path/ [;parameters][?query]#fragment
- 一些常见的URL规范化示例
  - URL协议名和主机名的小写化
    - HTTP://WWW.BAIDU.COM -> http://www.baidu.com
  - 删除Fragment（#）或多余的查询串，如?，&
    - http://www.example.com#seo -> http://www.example.com
  - 删除默认后缀或多余的www
    - http://www.example.com/index.html -> http://example.com

• DNS的基本概念和作用

- 域名系统（Domain Name System）
  - 将域名和IP地址相互映射的一个分布式数据库。
  - DNS地址解析可能成为重要瓶颈，甚至造成类似DOS的攻击效果



- **提升DNS性能的方法**

- 如何提高DNS解析模块的性能?
  - 并行 DNS Client
  - 缓存 cache DNS results
  - 预取 prefetch client

• 并行 DNS Client

- 并行的地址解析Client
  - 专门对付多个请求的并行处理
  - 容许一次发出多个解析请求
  - 协助在多个DNS server之间做负载分配
    - 例如，根据掌握的URL进行适当调度

- **缓存 cache DNS results**

- 增加DNS缓存的重要性
  - 面向海量URL和主机的搜索任务，如果没有DNS缓存，会造成频繁查询DNS服务器，从而造成类似于拒绝服务攻击（DOS）的副作用。
  - 针对小规模的网页搜索（如百万量级），可利用建立在内存中的DNS映射，既能加快网页信息的获取，后能降低对于DNS服务器的压力。
- DNS缓存的内容
  - Internet的DNS系统会定期刷新，交换更新的域名和IP信息
  - 缓存中有部分信息过期影响不大，但注意要适度刷新

- **预取 prefetch client**

- 为了减少等待查找涉及新主机的地址的时间，尽早将主机名投给DNS系统
- 与缓存系统的区别：缓存 – 用了才缓存；预取 – 不用也暂时先缓存
- DNS预取的基本步骤
  - 分析刚刚得到的网页
  - 从<href>属性中提取主机名
  - 向缓存服务器提交DNS解析请求
  - 结果存放在DNS缓存中（也可能用上，也可能用不上）

- **Sitemap，站点地图**

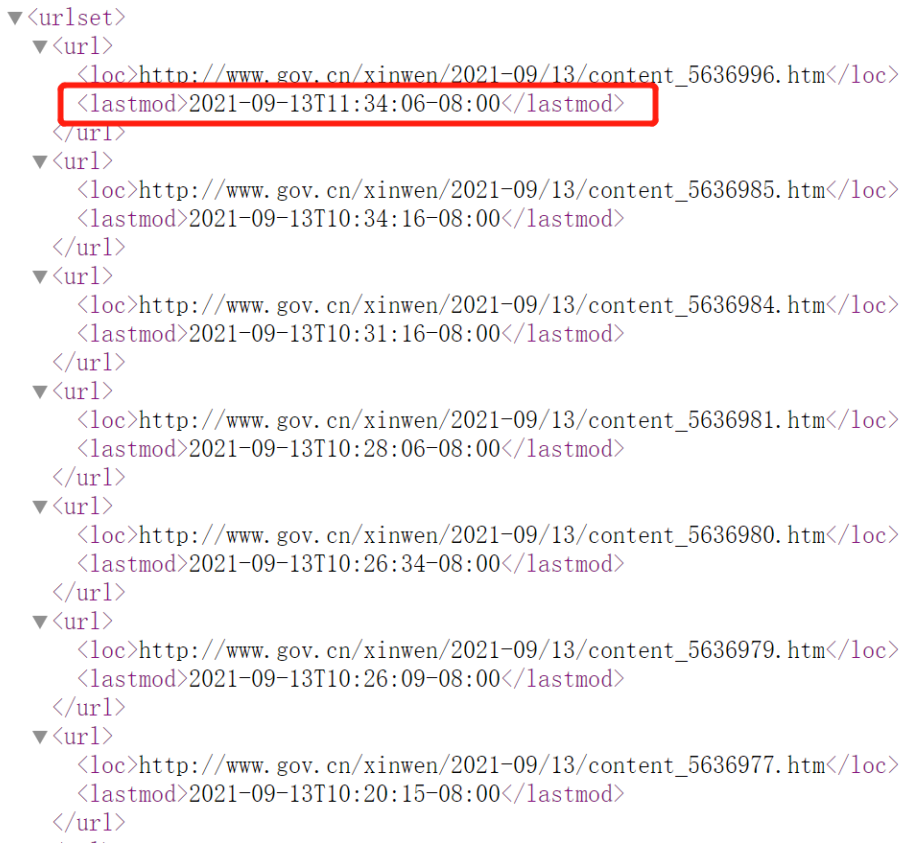
- 放在服务器根目录中的sitemap.xml，为爬虫指明抓取的建议
  - 协助爬虫找到隐藏的网页（如需要查询或表单才能够访问的内容）
  - Changefreq属性还提供有关更新频率的说明，以协助保障时新性

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>http://www.company.com/</loc>
    <lastmod>2008-01-15</lastmod>
    <changefreq>monthly</changefreq>
    <priority>0.7</priority>
  </url>
  <url>
    <loc>http://www.company.com/items?item=truck</loc>
    <changefreq>weekly</changefreq>
  </url>
  <url>
    <loc>http://www.company.com/items?item=bicycle</loc>
    <changefreq>daily</changefreq>
  </url>
</urlset>
```

Sitemap

- **Sitemap，站点地图**

- 案例：中央政府门户网站的站点地图-百度版本（http://www.gov.cn/baidu.xml）



- **Robots Exclusion，排斥协议**

- Sitemap是允许协议，而Robots是排斥协议
- 在服务器文档根目录中的文件robots.txt，包含一个路径前缀表，描述了服务器给出的抓取限制
- 例如，腾讯新闻的Robots.txt（news.qq.com/robots.txt）

```
User-agent: *
Disallow:
Sitemap: //www.qq.com/sitemap_index.xml
Sitemap: http://news.qq.com/topic_sitemap.xml
```

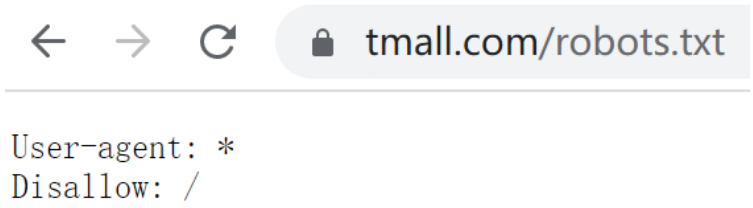
- **Robots Exclusion，排斥协议**

- 案例：中央政府门户网站的Robots.txt（www.gov.cn/robots.txt）

```
User-agent: *
Allow: /1
Sitemap:http://www.gov.cn/baidu.xml
Sitemap:http://www.gov.cn/google.xml
Sitemap:http://www.gov.cn/bing.xml
Sitemap:http://www.gov.cn/sogou.xml
Disallow:/2016gov/
Disallow:/2016shuju/
Disallow:/2016guoqing/
Disallow:/2016zhengce/
Disallow:/2016hudong/
Disallow:/2016fuwu/
Disallow:/2016xinwen/
Disallow:/premier/
Disallow:/2016guowuyuan/
Disallow:/2016public/
Disallow:/2016ducha/
Disallow:/guowuyuan/yangjing/
Disallow:/guowuyuan/yj.htm
Disallow:/c16629/gwyld_yj.htm
Disallow:/guowuyuan/yj_hy.htm
Disallow:/guoqing/2013-03/16/content_2583121.htm
Disallow:/guowuyuan/2015-12/24/content_5027563.htm
```

- **Robots Exclusion，排斥协议**

- 该限制只针对爬虫，一般浏览不受影响
- 该限制属于“君子协定”，没有强制执行力
  - 但目前，绝大多数搜索引擎都遵守该限制



淘宝商城_天猫
https://www.tmall.com/ - 2909条评价
由于该网站的robots.txt文件存在限制指令（限制搜索引擎抓取），系统无法提供该页面的内容描述 - 了解详情

爱淘宝-淘宝网购物分享平台
https://ai.taobao.com/ - 305条评价
由于该网站的robots.txt文件存在限制指令（限制搜索引擎抓取），系统无法提供该页面的内容描述 - 了解详情

淘宝网 - 淘!我喜欢
mai.taobao.com/ - 8483条评价
由于该网站的robots.txt文件存在限制指令（限制搜索引擎抓取），系统无法提供该页面的内容描述 - 了解详情

本节目录

- 网络爬虫的定义与需求
- 爬虫的基本要素
- **面向API的新爬虫任务**
- 常见的爬虫算法
- 常见反爬虫机制与应对策略

API

- **API的基本概念**

- 应用程序接口（Application Programming Interface）
- 2000年，Salesforce和eBay推出了自己的API，程序员可以访问并下载一些公开数据。从那时起，许多网站都提供API用于访问公共数据库。
- 通过开放的API 来吸引更多的用户和更多的创意，与具备分享、标准、去中心化、开放、模块化的Web 2.0时代相得益彰，为创造者和平台都带来价值。
- 同时，网页API也为开发人员提供了一种更友好的网络爬虫方式：
  - 直接获取资源列表，清晰、明确
  - 接收JSON或XML等格式化数据的反馈

API

- **基于API的数据爬取示例**

```
In [28]: import urllib.request as request
         import json
         url = 'https://api.douban.com/v2/movie/top250'
         crawl_content = request.urlopen(url).read()
         top20 = json.loads(crawl_content.decode('utf8'))['subjects']
         for movie in top20:
             url = 'https://api.douban.com/v2/movie/' + movie['id']
             movieContent = request.urlopen(url).read()
             print(json.loads(movieContent.decode('utf8')))
```


{'rating': {'max': 10, 'average': '9.6', 'numRaters': 947593, 'min': 0}, 'author': [{'name': '弗兰克·德拉邦特 Frank Darabont'}], 'alt_title': '肖申克的救赎 / 月黑高飞(港)', 'image': 'https://img3.doubanio.com/view/photo/s_ratio_poster/public/p480747492.jpg', 'title': 'The Shawshank Redemption', 'summary': '20世纪40年代末，小有成就的青年银行家安迪（蒂姆·罗宾斯 Tim Robbins 饰）因涉嫌杀害妻子及她的情人而锒铛入狱。在这座名为肖申克的监狱内，希望似乎虚无缥缈，终身监禁的惩罚无疑注定了安迪接下来灰暗绝望的人生。未过多久，安迪尝试接近囚犯中颇有声望的瑞德（摩根·弗里曼 Morgan Freeman 饰），请求对方帮自己搞来小锤子。以此为契机，二人逐渐熟稔，安迪也仿佛在鱼龙混杂、罪恶横生、黑白混淆的牢狱中找到属于自己的求生之道。他利用自身的专业知识，帮助监狱管理层逃税、洗黑钱，同时凭借与瑞德的交往在犯人中间也渐渐受到礼遇。表面看来，他已如瑞德那样对那堵高墙从憎恨转变为处之泰然，但是对自由的渴望仍促使他朝着心中的希望和目标前进。而关于其罪行的真相，似乎更使这一切朝前推进了一步……\n本片根据著名作家斯蒂芬·金（Stephen Edwin King）的原著改编。', 'attrs': {'pubdate': ['1994-09-10(多伦多电影节)', '1994-10-14(美国)'], 'language': ['英语'], 'title': ['The Shawshank Redemption'], 'country': ['美国'], 'writer': ['弗兰克·德拉邦特 Frank Darabont', '斯蒂芬·金 Stephen King'], 'director': ['弗兰克·德拉邦特 Frank Darabont'], 'cast': ['蒂姆·罗宾斯 Tim Robbins', '摩根·弗里曼 Morgan Freeman', '鲍勃·冈顿 Bob Gunton', '威廉姆·赛德勒 William Sadler', '克兰西·布朗 Clancy Brown', '吉尔·贝罗斯 Gil Bellows', '马克·罗斯顿 Mark Rolston', '詹姆斯·惠特摩 James Whitmore', '杰弗里·德曼 Jeffrey DeMunn', '拉里·布兰登伯格 Larry Brandenburg', '尼尔·吉恩托利 Neil Giuntoli', '布赖恩·利比 Brian Libby', '大卫·普罗瓦尔 David Proval', '约瑟夫·劳格诺 Joseph Ragno', '祖德·塞克利拉 Jude Ciccolella'], 'movie_duration': ['142 分钟'], 'year': ['1994'], 'movie_type': ['犯罪', '剧情']}, 'id': 'https://api.douban.com/movie/1292052', 'mobile_link': 'https://m.douban.com/movie/subject/1292052/', 'alt': 'https://movie.douban.com/movie/1292052', 'tags': [{'count': 194779, 'name': '经典'}, {'count': 165432, 'name': '励志'}, {'count': 147828, 'name': '信念'}, {'count': 133384, 'name': '自由'}, {'count': 98525, 'name': '美国'}, {'count': 94751, 'name': '人性'}, {'count': 73431, 'name': '人生'}, {'count': 60696, 'name': '剧情'}]}

基于豆瓣提供的API，获得评分最高的20部电影信息

- **基于API的数据爬取一般流程**

其中，第3-4步可能存在循环，即通过获得某个资源ID对应的详细信息，获得更多其他资源ID，并继续抓取更多信息。

API

• Token的概念、作用与取得方式

- API的调用需要用户身份认证（用户授权），而Token相当于授权后的通行证
- 使用Token认证而非用户名/密码认证方式的优点
  - Token 的生成完全独立于帐号密码，往来通信不会影响账号安全
  - 即使 Token 丢失或泄露，只需登录后台删除该Token即可

目前微博的API认证授权机制

XML

- **结构化数据反馈（1）：XML**

- 可扩展标记语言（Extensible Markup Language）
  - 是一种允许用户对自己的标记语言进行定义的源语言。
  - 提供统一的方法来描述和交换独立于应用程序的结构化数据。
- XML相比于HTML的优势所在：
  - 可扩展性方面：HTML不允许用户自行定义标识或属性，而在XML中，用户能够根据需要自行定义新的标识及属性名
  - 结构性方面：HTML不支持深层的结构描述，XML的文件结构嵌套可以复杂到任意程度，能表示面向对象的等级层次。
  - 可校验性方面：HTML没有提供规范文件以支持结构校验，而XML文件可以包括一个语法描述，使应用程序可以对此进行结构校验。

- **XML文件的实例与基本要求**

```
<?xml version="1.0" encoding="UTF-8"?>
- <item>
  - <venue>
      <address_1>162 Winn St</address_1>
      <state>MA</state>
      <zip>01803</zip>
      <lat>42.504240</lat>
      <repinned>False</repinned>
      <name>American Legion Hall</name>
      <city>Burlington</city>
      <id>486621</id>
      <country>us</country>
      <lon>-71.185790</lon>
    </venue>
  - <fee>
      <label>Price</label>
      <accepts>amazon</accepts>
      <currency>USD</currency>
      <description>per person</description>
      <amount>10.0</amount>
      <required>0</required>
    </fee>
    <status>past</status>
    <description><b>Looks like the storm predicte
      confirm the band. More details will follow.
      holidays with old friends and new at a spe
      American Legion Hall in Burlington (right c
      be a cash bar. You are welcome to invite fa
      done so on your reply, it will help me keep
      on your reply, &amp;quot;paying by check&amp;quo
      soon as you know you can attend. If you h
      we had a gift exchange which was alot of f
      receiving yourself. <br />Hope it will be a
    <how_to_find_us>We have rented the hall...so
```

必须有XML声明语句作为文档开头

良好的XML文档有且只有一个根节点

例如，在本实例中为item

可根据需要嵌套文件结构

例如，<venue>中包含<city>信息

所有的标记必须有相应的结束标记

例如，<zip>与</zip>的成对出现

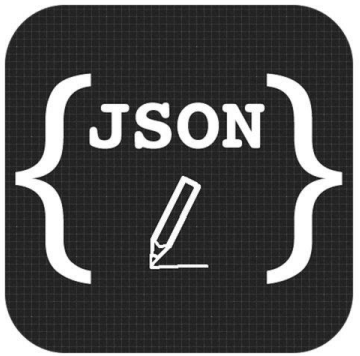
注意，所有的空标记也必须被关闭

XML

- **结构化数据反馈（2）：JSON**

- JS 对象表示法（JavaScript Object Notation）
  - 轻量级的数据交换格式，采用完全独立于编程语言的文本格式。
  - 简洁和清晰的层次结构使得 JSON 成为理想的数据存储和交换语言。
  - 易于人阅读和编写，同时也易于机器解析，并有效地提升网络传输效率。

```
"name":"Michael",
"address":
{
    "city":"Beijing",
    "street":" Chaoyang Road ",
    "postcode":100025
}
```

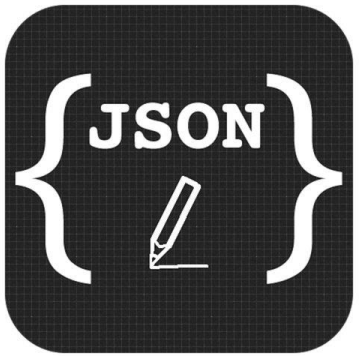


XML

- **JSON与XML的区别**

- XML的优点：格式统一、标准，容易与其他系统进行远程交互，方便共享
- XML的缺点：文件庞大，文件格式复杂，解析复杂且方式繁多，工作量大

- JSON的优点
  - 数据格式比较简单，易于读写，格式经过压缩，占用带宽小；
  - 支持多种语言，易于解析，可以简单的进行数据的读取；
  - 能直接为服务器端代码使用，大大简化了开发和维护工作。



XML

- **JSON的基本元素**

```xml
<?xml version="1.0" encoding="utf-8" ?>
<root>
    <resultcode>200</resultcode>
    <reason>Return Successd!</reason>
    <result>
        <area>江苏省苏州市</area>
        <location>电信</location>
    </result>
</root>
```

XML与JSON格式对比

```json
{
    "resultcode":"200",
    "reason":"Return Successd!",
    "result":{
        "area":"江苏省苏州市",
        "location":"电信"
    }
}
```

- JSON中的六大构造字符
- “[”　“]”表示数组开始/结束
- “{”　“}”表示对象开始/结束
- “：”表示名称的分隔符，“，”表示值的分隔符
- 从内容表示上看，JSON更为轻量，但在无缩进情况下层次化解析相对困难

本节目录

- 网络爬虫的定义与需求
- 爬虫的基本要素
- 面向API的新爬虫任务
- **常见的爬虫算法**
- 常见反爬虫机制与应对策略

- **最最基础的算法**

```
PROCEDURE SPIDER 1 (G)
        Let ROOT := any URL from G
        Initialize STACK <stack data structure>
        Let STACK := push(ROOT, STACK)
        Initialize COLLECTION <big file of URL-page pairs>

While STACK is not empty,
        URLcurr := pop(STACK)
        PAGE := look-up(URLcurr )
        STORE(<URLcurr , PAGE>, COLLECTION)
        For every URLi in PAGE,
                push(URLi , STACK)
Return COLLECTION
```

潜在风险：

重复收集的问题?

回路或不连通的解决方法?

如何控制搜集特定的一部分?

- **最最基础的改进算法**

```
PROCEDURE SPIDER(G, {SEEDS})
    Initialize COLLECTION <big file of URL-page pairs>
    Initialize VISITED <big hash-table>
    For every ROOT in SEEDS
        Initialize STACK <stack data structure>
        Let STACK := push(ROOT, STACK)
        While STACK is not empty,
            Do URLcurr := pop(STACK)
                Until URLcurr is not in COLLECTION
            insert-hash(URLcurr , VISITED)
            PAGE := look-up(URLcurr )
            STORE(<URLcurr , PAGE>, COLLECTION)
            For every URLi in PAGE,
                push(URLi , STACK)
Return COLLECTION
```

解决方案：

利用BigTable排除重复部分

基于种子集合控制爬取内容

通过更换种子重启动解决回路等

- **重复性的监测问题**

- 不仅URL本身可能重复，即使不同URL，也可能导致相似的文档内容
- 完全重复文档的检测方法
  - 检验和（Checksumming），所有字节进行加和，相同文档有相同检验和
  - 缺陷：相同检验和不代表相同（如打乱顺序）
- 近似重复文档的检测方法：指纹表示法
  - 1）对文档进行分词处理，并进行n-gram组合
  - 2）挑选部分n-gram用于表示这一文档
  - 3）对被选中的n-gram进行散列，以提升检索效率和减少指纹大小
  - 4）存储散列值作为文档指纹，通常存储在倒排索引中

- **重复性的监测问题**

Tropical fish include fish found in tropical environments around the world, including both freshwater and salt water species.

a) 原始文本

tropical fish include, fish include fish, include fish found, fish found in, found in tropical, in tropical environments, tropical environments around, environments around the, around the world, the world including, world including both, including both freshwater, both freshwater and, freshwater and salt, and salt water, salt water species

b) 3-gram

938 664 463 822 492 798 78 969 143 236 913 908 694 553 870 779

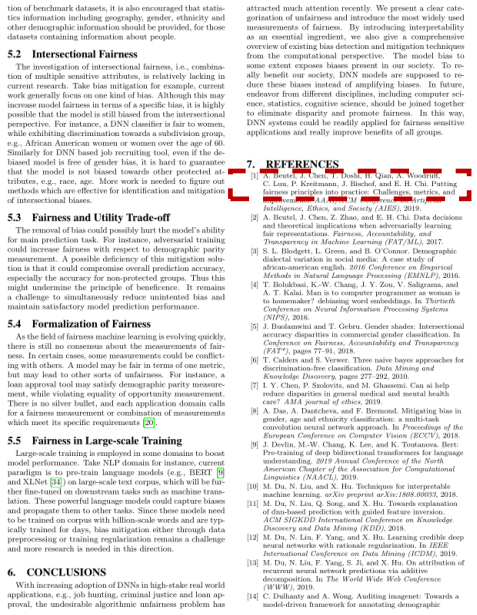
c) 散列值

664 492 236 908

d) 使用 0 mod 4 选择的散列值

图3-14 指纹生成过程实例

- **更进一步的遍历策略，从某博士读论文开始说起**



tion of benchmark datasets, it is also encouraged that statistics information including geography, gender, ethnicity and other demographic information should be provided, for those datasets containing information about people.

5.2 Intersectional Fairness

The investigation of intersectional fairness, i.e., combination of multiple sensitive attributes, is relatively lacking in current research. Take bias mitigation for example, current work generally focus on one kind of bias. Although this may increase model fairness in terms of a specific bias, it is highly possible that the model is still biased from the intersectional perspective. For instance, a DNN classifier is fair to women, while exhibiting discrimination towards a subdivision group, e.g., African American women or women over the age of 60. Similarly for DNN based job recruiting tool, even if the debiased model is free of gender bias, it is hard to guarantee that the model is not biased towards other protected attributes, e.g., race, age. More work is needed to figure out methods which are effective for identification and mitigation of intersectional biases.

5.3 Fairness and Utility Trade-off

The removal of bias could possibly hurt the model's ability for main prediction task. For instance, adversarial training could increase fairness with respect to demographic parity measurement. A possible deficiency of this mitigation solution is that it could compromise overall prediction accuracy, especially the accuracy for non-protected groups. Thus this might undermine the principle of beneficence. It remains a challenge to simultaneously reduce unintented bias and maintain satisfactory model prediction performance.

5.4 Formalization of Fairness

As the field of fairness machine learning is evolving quickly, there is still no consensus about the measurements of fairness. In certain cases, some measurements could be conflicting with others. A model may be fair in terms of one metric, but may lead to other sorts of unfairness. For instance, a loan approval tool may satisfy demographic parity measurement, while violating equality of opportunity measurement. There is no silver bullet, and each application domain calls for a fairness measurement or combination of measurements which meet its specific requirements [20].

5.5 Fairness in Large-scale Training

Large-scale training is employed in some domains to boost model performance. Take NLP domain for instance, current paradigm is to pre-train language models (e.g., BERT [9] and XLNet [34]) on large-scale text corpus, which will be further fine-tuned on downstream tasks such as machine translation. These powerful language models could capture biases and propagate them to other tasks. Since these models need to be trained on corpus with billion-scale words and are typically trained for days, bias mitigation either through data preprocessing or training regularization remains a challenge and more research is needed in this direction.

6. CONCLUSIONS

With increasing adoption of DNNs in high-stake real world applications, e.g., job hunting, criminal justice and loan approval, the undesirable algorithmic unfairness problem has attracted much attention recently. We present a clear categorization of unfairness and introduce the most widely used measurements of fairness. By introducing interpretability as an essential ingredient, we also give a comprehensive overview of existing bias detection and mitigation techniques from the computational perspective. The model bias to some extent exposes biases present in our society. To really benefit our society, DNN models are supposed to reduce these biases instead of amplifying biases. In future, endeavor from different disciplines, including computer science, statistics, cognitive science, should be joined together to eliminate disparity and promote fairness. In this way, DNN systems could be readily applied for fairness sensitive applications and really improve benefits of all groups.

7. REFERENCES

[1] A. Beutel, J. Chen, T. Doshi, H. Qian, A. Woodruff, C. Luu, P. Kreitmann, J. Bischof, and E. H. Chi. Putting fairness principles into practice: Challenges, metrics, and [illegible] *AAAI/ACM Conf. on AI, Ethics, and Society*, 2019.

Intelligence, Ethics, and Society (AIES), 2019.

[2] A. Beutel, J. Chen, Z. Zhao, and E. H. Chi. Data decisions and theoretical implications when adversarially learning fair representations. *Fairness, Accountability, and Transparency in Machine Learning (FAT/ML)*, 2017.

[3] S. L. Blodgett, L. Green, and B. O'Connor. Demographic dialectal variation in social media: A case study of african-american english. *2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

[4] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Thirtieth Conference on Neural Information Processing Systems (NIPS)*, 2016.

[5] J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency (FAT*)*, pages 77–91, 2018.

[6] T. Calders and S. Verwer. Three naive bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, pages 277–292, 2010.

[7] I. Y. Chen, P. Szolovits, and M. Ghassemi. Can ai help reduce disparities in general medical and mental health care? *AMA journal of ethics*, 2019.

[8] A. Das, A. Dantcheva, and F. Bremond. Mitigating bias in gender, age and ethnicity classification: a multi-task convolution neural network approach. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

[10] M. Du, N. Liu, and X. Hu. Techniques for interpretable machine learning. *arXiv preprint arXiv:1808.00033*, 2018.

[11] M. Du, N. Liu, Q. Song, and X. Hu. Towards explanation of dnn-based prediction with guided feature inversion. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2018.

[12] M. Du, N. Liu, F. Yang, and X. Hu. Learning credible deep neural networks with rationale regularization. In *IEEE International Conference on Data Mining (ICDM)*, 2019.

[13] M. Du, N. Liu, F. Yang, S. Ji, and X. Hu. On attribution of recurrent neural network predictions via additive decomposition. In *The World Wide Web Conference (WWW)*, 2019.

[14] C. Dulhanty and A. Wong. Auditing imagenet: Towards a model-driven framework for annotating demographic

[2] Y. Bao, H. Fang, and J. Zhang. Topicmf: Simultaneously exploiting ratings and reviews for recommendation. In *AAAI*, pages 2–8, 2014.

[3] Y. Cao, J. Li, X. Guo, S. Bai, H. Ji, and J. Tang. Name list only? target entity disambiguation in short texts. In *EMNLP*, pages 654–664, 2015.

[4] X. Chen, Z. Qin, Y. Zhang, and T. Xu. Learning to rank features for recommendation over multiple categories. In *Proceedings of the 39th international ACM SIGIR conference on Research & development in information retrieval*. ACM, 2016.

[5] X. Chen, F. Wang, Z. Qin, and Y. Zhang. Hlhpr: A hybrid local bayesian personal ranking method. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 21–22. International World Wide Web Conferences Steering Committee, 2016.

[6] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[7] P. Covington, J. Adams, and E. Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 191–198. ACM, 2016.

[8] Q. Cui, S. Wu, Q. Liu, and L. Wang. A visual and textual recurrent neural network for sequential prediction. *arXiv preprint arXiv:1611.06668*, 2016.

[9] Q. Diao, M. Qiu, C.-Y. Wu, A. J. Smola, J. Jiang, and C. Wang. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 193–202. ACM, 2014.

[10] N. Ejaz, T. B. Tariq, and S. W. Baik. Adaptive key frame extraction for video summarization using an aggregation mechanism. *Journal of Visual Communication and Image Representation*, 23(7):1031–1040, 2012.

[11] G. Gama, N. Elhadad, and A. Marian. Beyond the stars: Improving rating predictions using review text content. In *WebDB*, volume 9, pages 1–6. Citeseer, 2009.

[12] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Aistats*, volume 15, page 275, 2011.

[13] Y. Gong and X. Liu. Video summarization using singular value decomposition. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 174–180. IEEE, 2000.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[15] R. He and J. McAuley. Vbpr: visual bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1510.01784*, 2015.

[16] X. He, T. Chen, M.-Y. Kan, and X. Chen. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1661–1670. ACM, 2015.

[17] X. He, M. Gao, M.-Y. Kan, and D. Wang. Birank: Towards ranking on bipartite graphs. *IEEE Transactions on Knowledge and Data Engineering*, 29(3):57–71, 2017.

[18] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference Companion on World Wide Web (WWW)*, pages 173–182. International World Wide Web Conferences Steering Committee, 2017.

[19] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 549–558. ACM, 2016.

[20] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[21] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[22] R. M. Jiang, A. H. Sadka, and D. Crookes. Advances in video summarization and skimming. In *Recent Advances in Multimedia Signal Processing and Communications*, pages 27–50. Springer, 2009.

[23] W. Jiang, C. Cotton, and A. C. Loui. Automatic consumer video summarization by audio and visual analysis. In *ICME*, pages 1–6, 2011.

[24] A. Khosla, R. Hamid, C.-J. Lin, and N. Sundaresan. Large-scale video summarization using web-image priors. In *CVPR*, pages 2698–2705, 2013.

[25] G. Kim, L. Sigal, and E. P. Xing. Joint summarization of large-scale collections of web images and videos for storyline reconstruction. In *CVPR*, 2014.

[26] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *CVPR*, pages 1346–1353, 2012.

[27] L. Li, K. Zhou, G.-R. Xue, H. Zha, and Y. Yu. Video summarization via transferable structured learning. In *WWW*, pages 287–296. ACM, 2011.

[28] Y. Li, T. Zhang, and D. Tretter. An overview of video abstraction techniques. Technical report, Technical Report HPL-2001-191, HP Laboratory, 2001.

[29] H. Liu, J. He, T. Wang, W. Song, and X. Du. Combining user preferences and user opinions for accurate recommendation. *Electronic Commerce Research and Applications*, 12(1):14–23, 2013.

[30] Y.-F. Ma, X.-S. Hua, L. Lu, and H.-J. Zhang. A generic framework of user attention model and its application in video summarization. *IEEE transactions on multimedia*, 7(5):907–919, 2005.

[31] J. McAuley and J. Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172. ACM, 2013.

[32] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–52. ACM, 2015.

[33] E. Mendi and C. Bayrak. Shot boundary detection and key frame extraction using salient region detection and structural similarity. In *Proceedings of the 48th Annual Southeast Regional Conference*, pages 66–67, 2010.

[34] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[35] A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2007.

[36] A. Nagasaka and Y. Tanaka. Automatic video indexing and full-video search for object appearances. *Journal of Information Processing*, 15(2):316, 1992.

[37] J. Peng, Y. Zhao, and J. Qiu. Learning latent factor from review text and rating for recommendation. In *2015 7th International Conference on Modelling, Identification and Control (ICMIC)*, pages 1–6. IEEE, 2015.

[38] Z. Ren, S. Liang, P. Li, S. Wang, and M. de Rijke. Social collaborative viewpoint regression with explainable recommendations. *Under submission*, page 39, 2016.

[39] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pages 452–461. AUAI Press, 2009.

[40] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[41] Y. Tan, M. Zhang, Y. Liu, and S. Ma. Rating-boosted latent topics: Understanding users and items with ratings and reviews. In *IJCAI*. AAAI, 2016.

[42] D. Tang, B. Qin, and T. Liu. Learning semantic representations of users and products for document level sentiment classification. In *ACL*, pages 1014–1023, 2015.

[43] D. Tang, B. Qin, T. Liu, and Y. Yang. User modeling with neural network for review rating prediction. In *IJCAI*, pages 1340–1346, 2015.

[44] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015.

[45] H. Wang, N. Wang, and D.-Y. Yeung. Collaborative deep learning for recommender systems. In *SIGKDD*, pages 1235–1244, 2015.

[46] H. Wang, S. Xingjian, and D.-Y. Yeung. Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks. In *NIPS*, pages 415–423, 2016.

[47] M. Wang, R. Hong, G. Li, Z. J. Zha, S. Yan, and T.-S. Chua. Event driven web video summarization by tag localization and key-shot identification. *Multimedia, IEEE Transactions on*, 14(4):975–985, 2012.

[48] Z. Wang, M. Kumar, J. Luo, and B. Li. Sequence-kernel based sparse representation for amateur video summarization. In *Proceedings of the 2011 joint ACM workshop on Modeling and representing events*, pages 31–36, 2011.

[49] B. Wu, E. Zhong, B. Tan, A. Horner, and Q. Yang. Crowdsourced time-sync video tagging using temporal and personalized topic modeling. In *SIGKDD*, pages 721–730, 2014.

[50] C.-Y. Wu, A. Beutel, A. Ahmed, and A. J. Smola. Explaining reviews and ratings with paco: Poisson additive co-clustering. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 127–128. International World Wide Web Conferences Steering Committee, 2016.

[51] Y. Wu and M. Ester. Flame: A probabilistic model combining aspect based opinion mining and collaborative filtering. In *WSDM*, pages 199–208. ACM, 2015.

[52] Y. Xian, J. Li, C. Zhang, and Z. Liao. Video highlight shot extraction with time-sync comment. In *Workshop on Hot Topics in Planet-scale mobile computing and online Social networking*, pages 31–36, 2015.

[53] H. Xu, Y. Zhen, and H. Zha. Trailer generation via a point process-based visual attractiveness model. In *IJCAI*, pages 2198–2204, 2015.

[54] J. You, G. Liu, L. Sun, and H. Li. A multiple visual models based perceptive analysis framework for multilevel video summarization. *CSVT, IEEE Transactions on*, 17(3):273–285, 2007.

[55] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4694–4702, 2015.

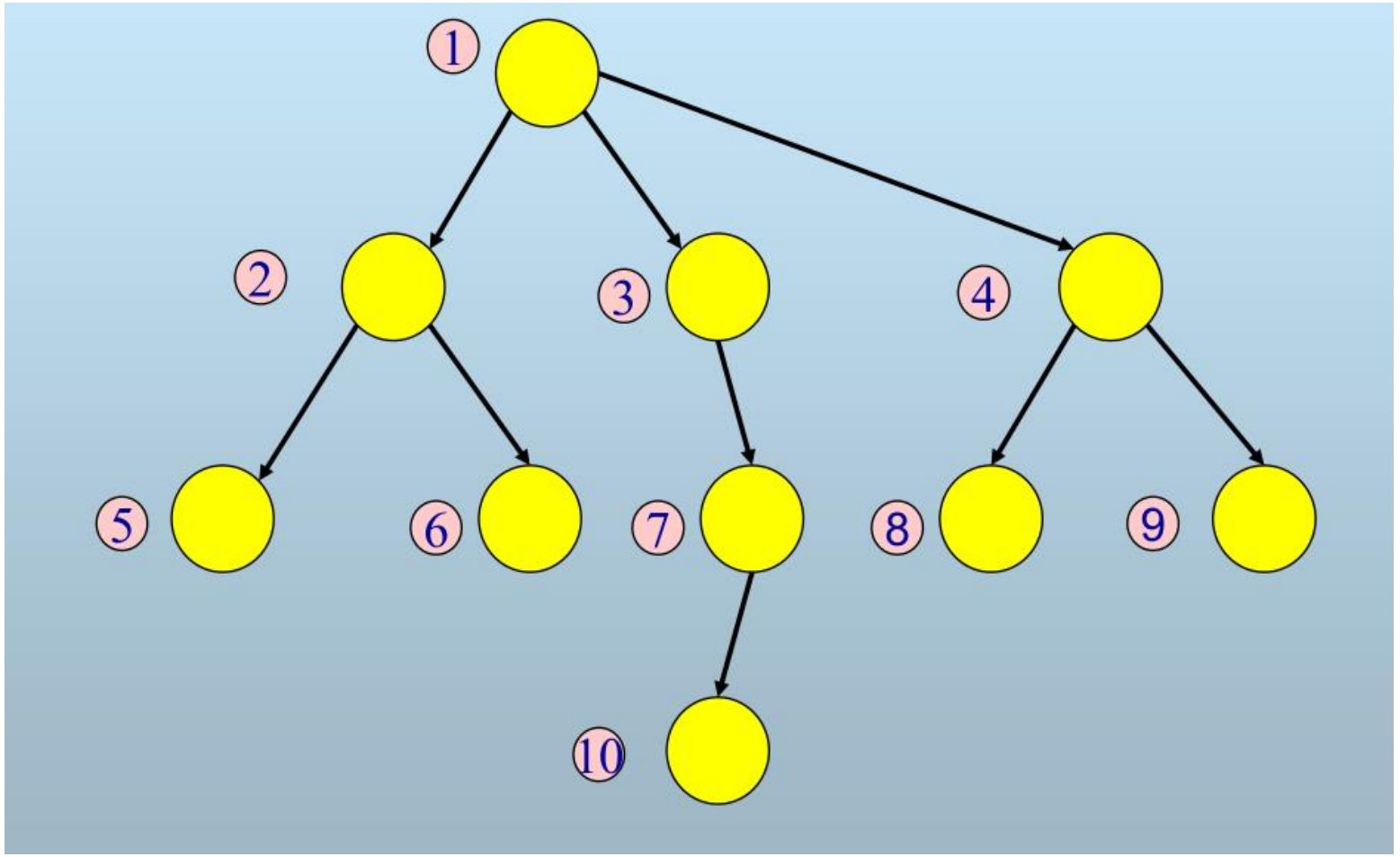
[56] W. Zhang, Q. Yuan, J. Han, and J. Wang. Collaborative multi-level embedding learning from reviews for rating prediction. In *IJCAI*. ACM, 2016.

[57] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 83–92. ACM, 2014.

[58] Y. Zhuang, Y. Rui, T. S. Huang, and S. Mehrotra. Adaptive key frame extraction using unsupervised clustering. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, volume 1, pages 866–870. IEEE, 1998.

- 遍历所有参考文献，然后从第一篇文献开始，遍历参考文献的参考文献

- **第一种遍历策略：广度优先算法**



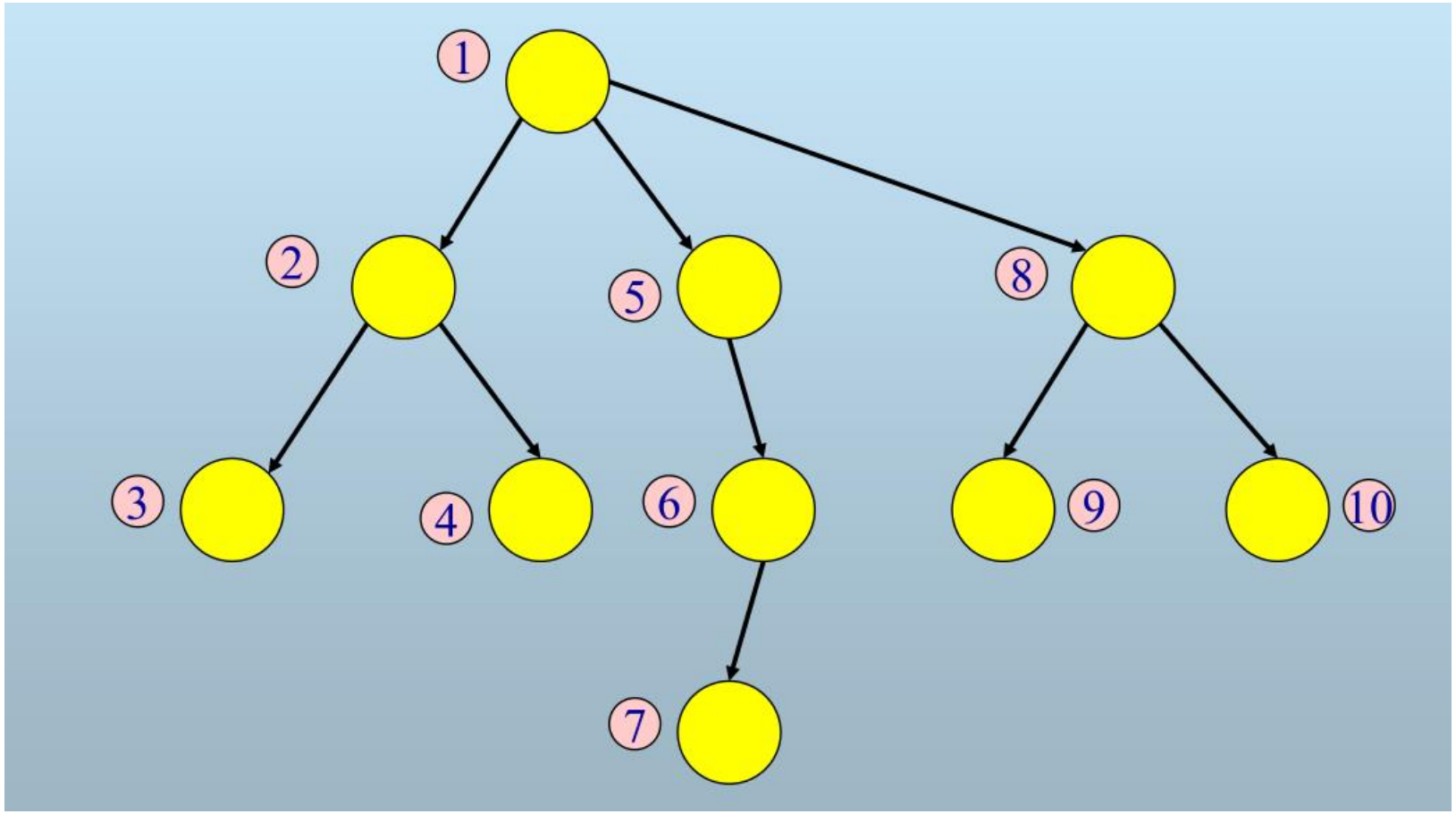
广度优先

• 第一种遍历策略：广度优先算法

```
PROCEDURE SPIDER(G, {SEEDS})
    Initialize COLLECTION <big file of URL-page pairs>// 结果存储
    Initialize VISITED <big hash-table>// 已访问URL 列表
For every ROOT in SEEDS
    Initialize QUEUE <queue data structure>// 待爬取URL 队列
    Let QUEUE := EnQueue(ROOT, QUEUE)
    While QUEUE is not empty,
            Do URLcurr := DeQueue(QUEUE)
                    Until URLcurr is not in VISITED
    insert-hash(URLcurr , VISITED)
    PAGE := look-up(URLcurr )// 爬取页面
    STORE(<URLcurr , PAGE>, COLLECTION)
    For every URL i in PAGE,// 链接提取
            EnQueue(URL, QUEUE)
Return COLLECTION
```

本质是维持一个URL队列，
先进先出

- **第二种遍历策略：深度优先算法**



深度优先

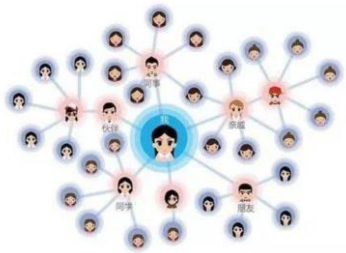
• 第二种遍历策略：深度优先算法

```
PROCEDURE SPIDER(G, {SEEDS})
    Initialize COLLECTION <big file of URL-page pairs>// 结果存储
    Initialize VISITED <big hash-table>// 已访问URL 列表
For every ROOT in SEEDS
    Initialize STACK <stack data structure>// 待爬取URL 栈
    Let STACK := push(ROOT, STACK)
    While STACK is not empty,
        Do URLcurr := pop(STACK)
            Until URLcurr is not in VISITED
        insert-hash(URLcurr , VISITED)
        PAGE := look-up(URLcurr )// 爬取页面
        STORE(<URLcurr , PAGE>, COLLECTION)
        For every URLi in PAGE,// 链接提取
            push(URLi , STACK)
Return COLLECTION
```

本质是维持一个URL栈，
先进后出

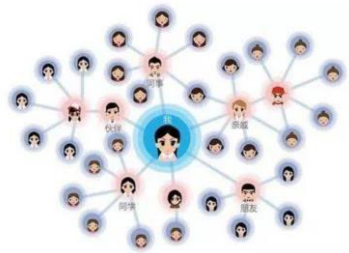
- **广度 or 深度优先算法?**

- 对于一般的网页爬虫而言，在时间无限的情况下，两种算法可以视作等价
- 假如在**有限步**条件下，两者的优劣如何?
  - 如果希望获得更为多样化的内容，应采用广度优先
  - 如果希望获得更为深层次的信息，应结合路径选择采用深度优先
    - 部分站点需要深层次浏览以判断站点信息质量



- **广度 or 深度优先算法?**

- 对于一般的网页爬虫而言，在时间无限的情况下，两种算法可以视作等价
- 此外，**不同的应用场景**，对两种算法的适应度可能不尽相同
  - 以社交网络场景为例，广度优先有着以下特点：
    - ✓ 通常而言，选择度数较高的起点，可以快速到达大量节点
    - ➢ 但全局性不佳，难以反映网络全貌
    - ➢ 另外，此类采样所得节点的度数往往虚高
    - ◆ Facebook网络，随机抽样平均度94，广度优先抽样竟达324



- **再进一步，从遍历要求到网页本身重要性的要求**

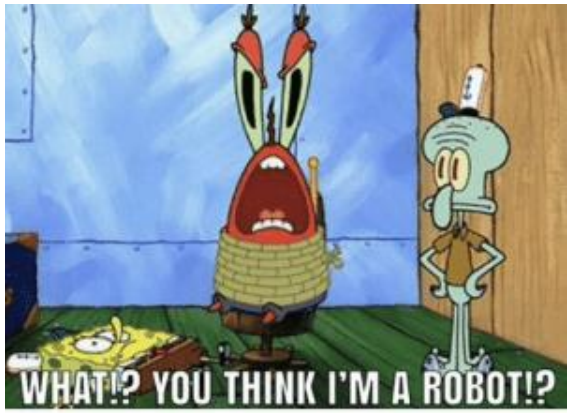
- 需要对网页的重要性进行评估，进而优先搜集重要的网页
- 根据经验，体现网页重要度的常见特征类别：
  - 启发式特征（简单统计指标）
    - 如，入度、父网页入度、镜像度、较小的目录深度（易于浏览）
  - 结构性特征
    - 如，PageRank/HIT值，各类Betweeness值等
  - 主题类特征（反应网页与特定需求的契合程度）

本节目录

- 网络爬虫的定义与需求
- 爬虫的基本要素
- 面向API的新爬虫任务
- 常见的爬虫算法
- **常见反爬虫机制与应对策略**

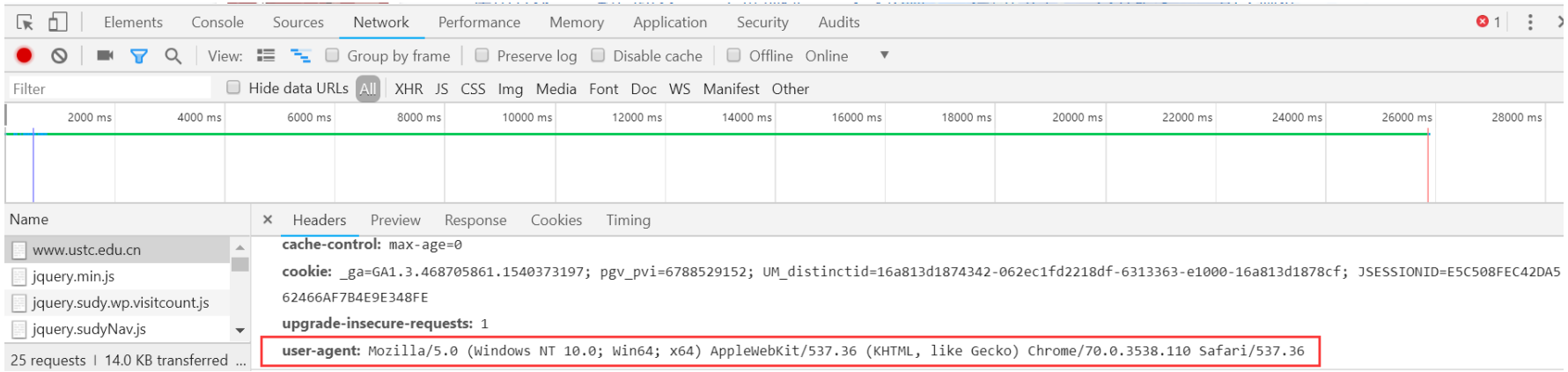
- **有爬虫，就有反爬虫**

- 使用任何技术手段，阻止别人批量获取自己网站信息
  - 顺便，大家也一起来当一下免费的数据标注员



- **常见反爬虫策略（1）： User Agent**

- 最常见的反爬虫策略，利用访问网站时浏览器发布的Request Headers信息中的User-Agent信息，判断用户使用何种方式浏览



UA.py

```
1  import requests
2
3  html = requests.get('https://zhihu.com').content
4  print(html.decode())
5
```

爬虫往往U-A部分为空

- **常见反爬虫策略（1）： User Agent**

- 应对策略：利用Python的Request库允许用户自定义请求头信息的手段
  - 在请求头信息中将 User-Agent 的值改为浏览器的请求头标识，从而绕开反爬虫机制

```
import requests
# 伪造请求头信息 欺骗服务器
headers = {"User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:9527.0) Gecko/20100101 Firefox/9527.0"}
resp = requests.get("http://127.0.0.1", headers=headers)
print(resp.status_code)
```

- **常见反爬虫策略（2）：IP/账号访问次数/频率**

- 通过限制特定IP地址/账号访问频率和次数进行反爬
  - 其本质在于判断浏览行为是否是**人类行为**

- 应对手段：
  - 构造 IP 代理池，然后每次访问时随机选择代理
    - Github中有相关服务，通过各种提供免费IP的网站来提供代理池
  - 每次爬取行为后间隔一段时间
  - 注册多个账号以保障数据收集
    - 挑战：账号本身的成本问题、账号被查封的危险

- **常见反爬虫策略（3）：验证码**

- 通过各类验证码，判断浏览者属于人类还是机器

从简单的字符识别到
复杂的逻辑推理

- 应对手段：
  - 简单的字符识别：基于机器学习与模式识别相关技术
  - 复杂的逻辑推理：人工辅助破解

- **常见反爬虫策略（4）：动态网页**

- 从网页的 url 加载网页的源代码之后，会在浏览器里执行JavaScript程序
  - 网页内容由脚本加载，而直接抓取则只能得到空白页面
  - 此类情况在抓取在线播放的音视频文件时尤为常见
- 应对手段：
  - 核心思路：模拟调用请求
  - 使用审查元素分析ajax请求，如此循环直到获得包含数据信息的json文件

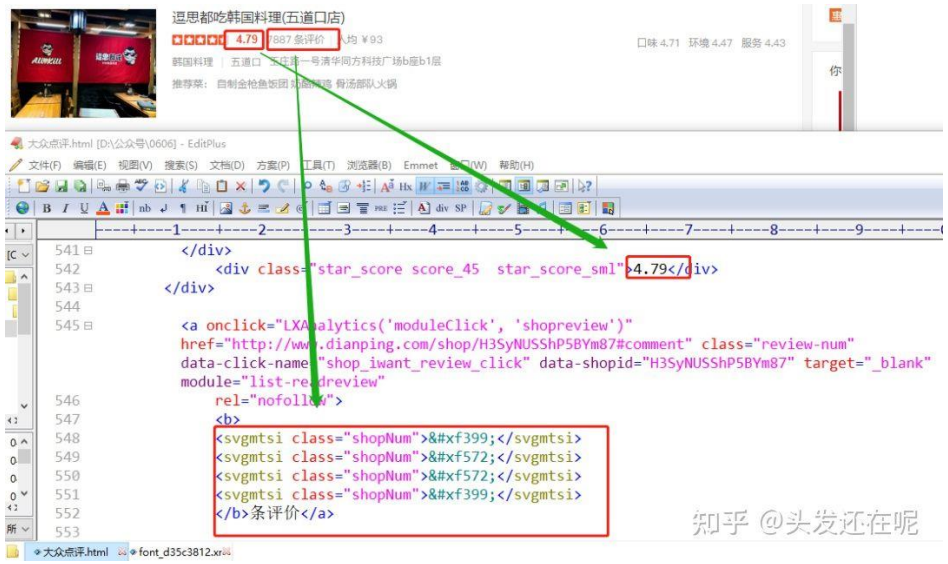
- **常见反爬虫策略（5）：蜜罐技术**

- 网页上会故意留下一些人类看不到或者绝对不会点击的链接。由于爬虫会从源代码中获取内容，所以爬虫可能会访问这样的链接
- 只要网站发现有IP访问这个链接，立刻永久封禁访问者，从而难以继续爬取

- 应对手段：
  - 核心思路：干涉爬虫路径
  - 通过工具库判断页面上的隐含元素，使爬虫避开这些元素，可以部分回避蜜罐的诱导。

• 常见反爬虫策略（6）：加密-解密技术

- 对网页中的关键信息进行加密或混淆，来增加
- 一类常见案例：页面上显示为正常文本，而源代码中为编号
  - 网页加载时需要借助字体库，将编码转为文字
    - 字体库/映射可以动态更新！
    - 解决方法：
      - ➢ 通过抓包获取字体库/映射
      - ➢ 基于映射进行解码



参考链接：https://zhuanlan.zhihu.com/p/406615464

- **常见反爬虫策略（7）：用户权限限制**

- 不同类型/级别的用户给予不同的内容权限
  - VIP、SVIP、蓝钻、红钻、绿钻、各种钻……

- 应对手段：
  - 氪金，就可以变强

- **其他的反爬虫策略**

- 不同的网页结构：每个相同类型的页面的源代码格式均不相同
- 同样也是双刃剑：增加爬取难度的同时降低用户浏览体验（逼死强迫症）

- 多模态的呈现方式：文字转为图像或视频
- 应对策略：OCR、语音识别、图像/视频标签技术

本章小结

网络爬虫

- 网络爬虫的定义与需求
- 爬虫的基本要素
- 面向API的爬虫任务
- 常见的爬虫算法
- 常见反爬虫机制与应对策略

tongxu@ustc.edu.cn