计算机类专业“互联网+”创新型精品教材

数据采集技术

主编　罗坤　薛龙　杨健

数据采集技术

主编　罗坤　薛龙　杨健

北京出版集团
北京出版社

北京出版集团
北京出版社

图书在版编目（CIP）数据

数据采集技术 / 罗坤，薛龙，杨健主编. —北京：北京出版社，2024.6

ISBN 978-7-200-18683-3

Ⅰ. ①数… Ⅱ. ①罗… ②薛… ③杨… Ⅲ. ①数据采集—高等职业教育—教材 Ⅳ. ① TP274

中国国家版本馆 CIP 数据核字（2024）第 105991 号

数据采集技术

SHUJU CAIJI JISHU

主　　编：罗坤　薛龙　杨健
出　　版：北京出版集团
　　北京出版社
地　　址：北京北三环中路 6 号
邮　　编：100120
网　　址：www.bph.com.cn
总 发 行：北京出版集团
经　　销：新华书店
印　　刷：定州启航印刷有限公司
版 印 次：2024 年 6 月第 1 版　2024 年 6 月第 1 次印刷
成品尺寸：185 毫米 ×260 毫米
印　　张：12.5
字　　数：281 千字
书　　号：ISBN 978-7-200-18683-3
定　　价：42.00 元

教材意见建议接收方式：010-58572341　邮箱：jiaocai@bphg.com.cn
如有印装质量问题，由本社负责调换
质量监督电话：010-82685218　010-58572341　010-58572393

目录

项目一

初识数据采集

在这个信息爆炸的时代，互联网上积累了大量数据，这些数据集中在一起形成了互联网大数据。对实时大数据进行分析，对于任何主体来讲，它的价值都不言而喻，特别是中小微公司无法通过自身产生大量的数据，而如果能够合理采集分析有价值的数据，就可以弥补自身的先天数据短板。通过爬虫爬取有价值的数据解决的就是数据采集问题，有效甚至高效、自动地采集数据是最基础的工作，也是最重要的工作。本项目主要介绍互联网数据的来源与特征、数据采集的概念与数据采集的方式方法以及网络爬虫的相关知识。

任务一　数据采集

任务描述

在信息技术快速发展的今天，数据采集已经被广泛地应用于各行各业。本任务我们学习互联网数据的相关知识及数据采集概念与技术框架。

任务目标

1. 会分析互联网数据的来源与特征。
2. 能自觉遵守数据采集相关法律法规。
3. 培养科技筑梦、强国有我的坚定信念。

任务实施

一、互联网数据

互联网大数据在大数据技术研究和应用中具有重要位置，由于互联网大数据的数据来源、数据类型和语义更加丰富，数据的开放性更好，数据的流动性更大，并且随着“互联网 +”国家战略的实施，各个行业与互联网之间的联系越来越密切，互联网大数据的价值体现也就更加广泛和多样。

（一）互联网数据来源

广义的互联网数据既包括各种互联网 Web 应用中不断累积产生出来的数据，也包括 Web 后台的传统业务处理系统产生的数据。狭义的互联网数据主要是指基于互联网 Web 应用所产生的数据，如新闻信息，微博、网络论坛帖子，电商评论等。

在互联网大数据研究和应用中，常见的数据来源有以下几种途径。

1. 百科知识库

大数据技术应用是一种基于经验数据的应用，经验数据的质量、完整性和可得性对于大数据的成功实施非常重要。但是，经验知识往往存在于每个人的大脑中，其表达、存储并不是很容易的事。随着互联网应用的扩展，出现了很多百科知识库，如百度百科、维基百科等。开放式的知识管理方式允许每个人对知识的正确性进行维护，因此出现了一些高质量的百科知识库，对于在大数据应用中进行知识获取、分析和推理具有重要价值。

2. 新闻网站

新闻信息是互联网大数据另一个重要的组成部分，涵盖社会新闻、科技新闻、国际新闻等。随着新闻发布机制的创新，互联网上新闻信息发布的及时性提高，一些个性化推送平台使得新闻的受众选择更加精准。各类新闻信息体现了当前各个领域的重要事件以及事件的演化过程，因此为大数据的动态性和深度分析挖掘提供了很好的数据源和示例。

3. 社交媒体 / 网络

微博、网络论坛等各种社交平台已经成为人们聊天、分享信息、交换意见的重要场合，不断地产生各种即时信息（User Generated Content，UGC）。这些数据体现了人们的观点、情绪、行为，以及群体关注的热点、话题等许多信息。这些信息已经逐步被越来越多的机构重视，用来进一步挖掘分析，为提升客户服务、产品质量提供准确资料。

社交网络主要来源于社交平台，它更侧重于人际关系数据，而社交媒体更侧重于内容，也有很多文献资料并不太区分社交网络和社交媒体。

4. 评论信息

股票评论、商品评论、酒店评论、服务质量评论等许多评论信息在互联网上广泛存在，它们属于典型的短文本，这类数据在大数据分析应用中具有典型的代表性，是一种重要的大数据。其分析和处理方法不同于新闻信息之类的长文本，互联网上的各类评论

信息为相应的技术研究和应用开发提供了充足的数据。

5. 位置型信息

随着移动互联网应用的快速普及，人们越来越习惯于在社交平台上进行签到，移动社交平台通常也记录了人们移动的位置和轨迹。这类数据作为一种重要的大数据来源，在大数据分析应用中具有较高价值，因此也是值得关注的互联网大数据之一。

此外还有很多其他途径来源的互联网大数据，这里就不一一列举了。

互联网数据的特征

（二）互联网数据的特征

互联网数据具有典型的大数据特征，即数据体量巨大、数据类型多样化、数据价值密度低、数据产生和处理速度快。除了具备这些基本特征外，互联网数据还有如下所述的一些新特征。

1. 数据类型和语义更加丰富

互联网数据除了传统的基本数据类型以外，还有文本型、音 / 视频、用户标签、地理位置信息、社交连接数据等。这些数据广泛存在于各类互联网应用中，例如新闻网站上的新闻、网络论坛中的帖子、基于位置服务系统（LBS）中的经纬度信息，以及微博中用户关注所形成的连接数据。

这种数据虽然本质上属于字符串、整型等基本数据类型，但是它们经过重新整合已经形成了具有一定语义的数据单元，如从用户评论文本中可以引申出用户的情感、人格，从用户的轨迹数据中可以引申出其活动规律，等等。

2. 数据的规范化程度更弱

弱规范性的数据是人们表达灵活性的体现，因此具有很高的研究价值，在以关系型数据为主的时代，此类数据并不多见。由于互联网数据的动态性、交互性都比较强，在信息传播作用下，用户生成的信息通常也有很大的相似性。此外，用户生成的信息是可以由用户控制的，也就是用户可以在此后进行修改、删除。因此，在采集互联网大数据时就可能会出现信息内容不一致的情况。

此外，互联网应用中对数据的校验并不是很严格，甚至可能是用户自定义的，这种数据规范化方式与 OLTP（联机事务处理）预先定义的模式也完全不同。典型的是微博中的用户标签，每个人可以根据自己的偏好设定自己的标签，两个不同的标签可能具有相同的含义，而相同的标签对不同用户来说可能有不同的含义。

3. 数据的流动性更大

在 OLTP 中，数据产生的速度取决于业务组织和规模，除了银行、电信等大型的联机系统外，OLTP 数据流动性一般并不高，数据生成速度也很有限。但是在互联网环境下，越来越多的应用由于面对整个互联网用户群体而使得数据产生、数据流动性极大增强，如微博、LBS 服务系统等，这种流动性主要体现在信息传播及数据在不同节点之间的快速传递。这种特点也就决定了大数据分析技术要具备对数据流的高速处理能力，挖掘算法要能够支持对数据流的分析，技术平台要具备充足的并行处理能力。

4. 数据的开放性更好

OLTP 具有很强的封闭性，但对于互联网大数据而言，由于互联网应用架构本身具有去中心化的特点，也就使得各种互联网应用中的数据在较大范围内是公开的，可以自由获取。而且由于互联网应用的开放性特点，对于用户的身份审查并不太严格，用户之间进行数据共享和自由分享也就变得更加容易。

5. 数据的来源更加丰富

随着智能终端的快速普及、通信网络的升级换代加速、智能技术和交互手段越来越丰富，互联网应用程序形式将变得丰富多彩，也将产生与以往不同的数据形式，如虚拟现实（VR）技术的应用就可能直接将人的真实表情数据、生理数据记录下来。此外，云计算、物联网技术的出现带来了新的服务模式，它们与互联网的结合也将极大地扩大互联网大数据来源。多种不同来源的数据以互联网为中心进行融合，正符合大数据的基本特征，因此可以在这个基础上进行更有效的分析和挖掘。

6. 价值体现形式更加多样化

随着“互联网 +”国家战略的推进，互联网思维在各个行业得到运用，互联网大数据与每个行业领域都存在结合点，因此大数据的价值体现也就不会仅局限于互联网应用自身。例如互联网与出租车的结合，使得基于互联网大数据的车流预测、路径规划更具有全局性。

互联网大数据与科学研究结合在一起也形成了目前颇具特色的研究范式。从以社会调查和试验为主要基础的社会科学领域，逐渐过渡到以互联网为背景来构建自己的数据源，例如很多的研究以微博、Twitter 中的用户行为数据为基础，开展一些心理、情感方面的研究，也凸显了互联网大数据价值的多样化。在新闻学、金融学、认知心理学、法学等众多领域，都体现了互联网大数据与各个学科领域结合的效用。

二、数据采集

（一）数据采集概念

数据采集（Data Acquisition，DAQ）又称数据获取，是指从各类数据库、机器设备、传感器等自动采集信息的过程。数据采集的对象在新一代数据体系中将传统数据体系中没有考虑过的新数据源进行了归纳与分类，将其分为线上行为数据与内容数据两大类。

（1）线上行为数据：页面数据、交互数据、表单数据、会话数据等。

（2）内容数据：应用日志、电子文档、机器数据、语音数据、社交媒体数据等。

随着互联网大数据时代的来临，数据采集面临着更多新的难题。传统数据与互联网大数据的数据采集的区别如表 1–1 所示。

表 1-1　传统数据与互联网大数据的数据采集的区别

传统数据	互联网大数据
来源单一	来源广泛

续表

传统数据	互联网大数据
结构单一	数据类型丰富，包括结构化、半结构化、非结构化
关系数据库和并行数据仓库	传统关系数据库、数据仓库、分布式数据库

从表 1－1 可以看出传统数据采集的不足：传统的数据采集来源单一，且存储、管理和分析的数据量也相对较小，大多采用关系数据库和并行数据仓库即可处理。而目前所处的互联网大数据时代的数据来源更广泛，类型更丰富，实时要求更高，这大幅度提高了数据采集的难度。

（二）数据采集的典型应用场景

1. 知识信息储备

服务、保险、汽车、维修、医药等行业需要储备规模巨大的资料库，而传统、庞大、繁杂的解答手册和知识系统会造成重复查询，导致系统延迟和成本上升，使用数据采集技术将有效缓解这类问题。例如，某全球航空制造商部署了 IBM InfoSphere Data Explore，使技师、支持人员和工程师能够通过单一访问点即时查看位于不同应用程序中的信息。部署第一年，该公司全天候支持的呼叫时间从过去的 50 分钟缩短为 15 分钟，每年节约 3600 万美元。这就是通过数据采集技术将众多数据库集中在一起所产生的价值。

2. 搜索技术

人们几乎每天都在使用搜索引擎。搜索引擎离不开爬虫，如百度搜索引擎的网络爬虫（简称爬虫）称为百度蜘蛛（Baiduspider）。百度蜘蛛每天会在海量的互联网信息中爬取优质信息并收录。当用户在百度搜索引擎上检索相应的关键词时，百度将对关键词进行分析处理，从收录的网页中找出相关网页，最后按照一定的排名规则进行排序并将结果展现给用户。在这个过程中，百度蜘蛛起到了至关重要的作用。那么，如何覆盖互联网中更多的优质网页，又如何筛选并去除重复的页面呢？这些都是由百度蜘蛛的算法决定的。采用不同的算法，爬虫的运行效率会有所不同，爬取结果也会有所差异。

除了百度搜索引擎外，其他搜索引擎也离不开爬虫，它们也拥有自己的爬虫。比如，360 的爬虫叫 360 Spider，搜狗的爬虫称 Sogou Spider，必应的爬虫称 Bingbot。

3. 其他网络爬虫应用

互联网上有着无数的网页，它们包含着海量的信息。网络爬虫可以代替手工做很多事情，除了可以作为搜索引擎，还可以爬取网站上面的图片等信息。例如，可以将某些网站上的图片全部爬取下来，集中进行浏览。同时，网络爬虫也可以用在金融投资领域，如可以自动爬取一些金融数据以进行投资分析等。除此之外，还有如下一些应用。

（1）新闻网站集中阅读。用户每次都要分别打开不同的新闻网站进行浏览，这样做比较麻烦。此时可以利用网络爬虫将这些新闻网站中的新闻信息爬取下来，再集中进行阅读。

（2）过滤广告。浏览网站时经常有广告出现，同样可以利用爬虫技术将对应网页上的有用信息爬取过来，这样就可以自动过滤掉这些广告，方便对信息进行阅读与使用。

（3）精准营销。如何找到目标客户及目标客户的联系方式是一个关键问题。可以手动在互联网中寻找，但是效率很低。而如果利用爬虫，便可以设置对应的规则，自动从互联网中采集目标用户的联系方式等数据，从而做到精准营销。

（4）网站用户信息分析。比如，分析某网站的用户活跃度、发言数、热门文章等信息。如果不是网站管理员，而是通过人工统计每页的数据，那么工作量将极其庞大。利用爬虫可以轻松地采集到这些数据，以便进行进一步的分析，而这一切爬取的操作都是自动进行的，只需要编写好对应的爬虫代码，并设计好对应的规则即可。

（三）数据采集步骤

在大数据价值链中，数据采集阶段的任务是以数字形式将信息聚合，以待存储和分析处理。数据采集过程可分为三个步骤，如图 1－1 所示。首先，数据收集（data collection），数据来源包括日志文件、传感器、Web 爬虫等；其次，数据传输（data transmission），包括物理层和网络层；最后，数据预处理（data preprocessing），包括数据整合、数据清洗和冗余消除等。数据传输和数据预处理没有严格的次序，数据预处理可以在数据传输之前或之后。

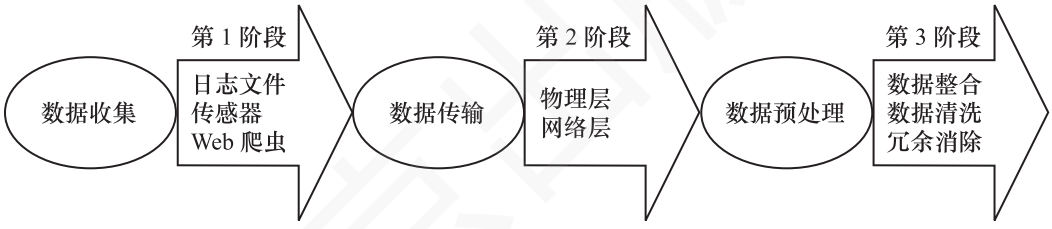


图 1-1　数据采集的一般步骤

（四）数据分类

按照数据的形态，可以把数据分为结构化数据和非结构化数据两种。

（1）结构化数据（如传统关系型数据库数据）的字段有固定的长度和语义，计算机程序可以直接处理。

（2）非结构化数据有文本数据、图像数据、自然语言数据等，计算机程序无法直接处理，需要进行格式转换或信息提取。

按照数据连接的方式，数据又可分为本地数据和网络数据等。

按照描述不同的实体，可以把数据分为类别数据和数值数据两种。

1. 类别（categorical）数据

（1）名义（nominal）数据：类别没有大小顺序的数据，如民族、性别、种族、颜色、院系、专业等。

（2）序数（ordinal）数据：类别有大小顺序的数据，如成绩等级、行业排名等。

2. 数值（numerical）数据

（1）离散（discrete）数据：是指其数值只能用自然数或整数单位计算的数据，如企业个数、职工人数、设备台数等。

（2）连续（continuous）数据：是指一定区间内可以任意取值的数据，其数值是连续不断的，相邻两个数值之间可做无限分割，即可取无限个数值，如身高、体重、里程等。

三、数据采集方式和方法

（一）数据采集方式

按照不同的视角，数据采集有不同的方式。

1. 主动 / 被动视角

按照数据采集的是主动还是被动视角，数据采集可分为推（push）方式和拉（pull）方式。

推方式的主动权在数据源系统方，数据源系统方根据自己数据产生的方式、频率以及数据量，采用一种适合数据源系统的方式将数据推送到数据处理系统，其特点是数据量、数据格式以及数据提供频率与数据生成方式相关。

拉方式的主动权则掌握在数据处理端，数据获取的频率、数据量和获取方式完全由数据处理端决定。

2. 即时性视角

按照数据采集的即时性视角，数据采集又可分为实时采集与离线采集。

实时采集是指在数据产生时立即对其进行处理和分析，并将结果传递到目标系统中。该方法通常用于需要快速响应和即时分析的场景，如金融交易、在线广告等。实时采集需要具备高速度、高可靠性和高扩展性等特点，以确保数据能够及时传输和处理。

离线采集是指将数据存储在本地或远程存储设备中，并在后续时间段内对其进行处理和分析。该方法通常用于需要大规模数据处理、长时间分析和历史数据回顾的场景，如机器学习、数据挖掘等。离线采集需要具备高容量、高效率和高灵活性等特点，以确保能够完成大规模数据的存储和分析。

（二）数据采集方法

数据采集的对象和来源多种多样，如传感器、系统日志、数据库和 Web 爬虫等，它们对应的数据采集方法也存在差异。下面介绍几种常见的数据来源及相应采集方法。

1. 传感器

传感器常用于测量物理环境变量并将其转化为可读的数字信号以待处理，根据测量类型的不同，分为压力、振动、位移、红外光、紫外光、温度、湿敏、离子、微生物等传感器。信息通过有线或无线网络传送到数据采集点。

有线传感器网络通过网线收集传感器的信息，这种方式适用于传感器易于部署和管理的场景。

无线传感器网络（wireless sensor network，WSN）利用无线网络作为信息传输的载体，适用于没有能量或通信的基础设施的场合。无线传感器网络通常由大量微小传感器节点构成，微小传感器由电池供电，被部署在应用指定的地点收集感知数据。当节点部署完成后，基站将发布网络配置 / 管理或收集命令，来自不同节点的感知数据将被汇集并转发到基站以待处理。基于传感器的数据采集系统被认为是一个信息物理系统。

2. 系统日志

日志由数据源系统产生，以特殊的文件格式记录系统的活动。几乎所有在数字设备上运行的应用的日志文件都非常有用。例如，Web 服务器通常要在日志文件中记录网站用户的单击、键盘输入、访问行为以及其他属性。

用于捕获用户在网站上的活动的 Web 服务器日志文件格式有 NCSA 通用日志文件格式、W3C 扩展日志文件格式和 Microsoft IIS 日志文件格式三种类型。数据库也可以用来替代文本文件存储日志信息，以提高海量日志的查询效率。在大数据领域，还可基于分布式的海量日志采集、聚合和传输系统 Flume 及支持高吞吐量的分布式发布 / 订阅消息系统进行日志采集。

3. 数据库

传统企业会使用传统的关系型数据库（如 MySQL 和 Oracle 等）来存储数据。随着大数据时代的到来，Redis、MongoDB 和 HBase 等 NoSQL 数据库（泛指非关系型数据库）逐渐在互联网企业中得到广泛使用。

数据库一般可通过应用程序编程接口（application programming interface，API）以主动或被动方式采集数据，采集策略可基于定时或者数据库触发机制增量获取或完整刷新等。独立的 ETL（extract－transform－load）技术可完整处理常见数据来源的采集、转换和处理，通过对数据进行提取、转换、加载，最终挖掘数据的潜在价值。

4. Web 爬虫

Web 爬虫（也称网络爬虫）是指从搜索引擎下载并存储网页的程序。Web 爬虫按顺序访问初始队列中的一组统一资源定位符（uniform resource locator，URL），并为所有 URL 分配一个优先级，然后从队列中获得具有一定优先级的 URL，下载该网页，随后解析网页中包含的所有 URL 并添加这些新的 URL 到队列中。这个过程一直重复，直到爬虫程序停止为止。Web 爬虫是网站应用（如搜索引擎）的主要数据采集方式之一。

Web 爬虫数据采集过程由选择策略、重访策略、礼貌策略以及并行策略决定。选择策略决定哪个网页将被访问；重访策略决定何时检查网页是否更新；礼貌策略防止过度访问网站；并行策略则用于协调分布的爬虫程序。

四、数据传输与预处理

（一）数据传输

原始数据采集后必须将其传送到数据存储基础设施（如数据中心）等待进一步处理。数据传输过程可以分为 IP 骨干网传输和数据中心传输两个阶段，如图 1－2 所示。

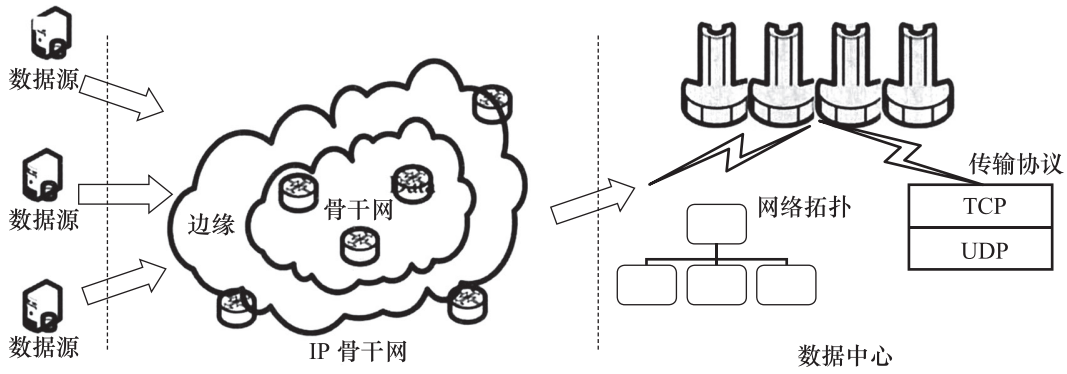


图 1-2 数据传输的一般过程和阶段

1. IP 骨干网传输

IP 骨干网提供高容量主干线路将大数据从数据源传递到数据中心。传输速率和容量取决于物理媒体和链路管理方法。

（1）物理媒体：通常由许多光缆合并在一起增加容量，并需要拥有多条路径以确保路径失效时能进行重路由。

（2）链路管理：决定信号如何在物理媒体上传输。过去 20 年间，IP over WDM 技术得到了深入研究。波分复用技术（wavelength division multiplexing，WDM）是在单根光纤上复用多个不同波长的光载波信号。为了解决电信号带宽的瓶颈问题，正交频分复用（orthogonal frequency division multiplexing，OFDM）被认为是未来的高速光传输技术的候选者。OFDM 允许单个子载波的频谱重叠，构建数据流更灵活、资源有效使用的光网络。

2. 数据中心传输

数据传递到数据中心后，将在数据中心内部进行存储位置的调整和其他处理，这个过程称为数据中心传输，涉及数据中心体系架构和传输协议。

（1）数据中心体系架构。数据中心由多个装备了若干服务器的机架构成，服务器通过数据中心内部网络连接。许多数据中心基于权威的 2 层或 3 层 fat－tree 结构的商用交换机构建。一些其他的拓扑结构也用于构建更为高效的数据中心网络。由于电子交换机的固有缺陷，使它在增加通信带宽的同时减少能量消耗变得非常困难。数据中心网络中的光互联技术能够提高吞吐量、降低延迟和减少能量消耗，被认为是有前景的解决方案。

（2）传输协议。TCP 和 UDP 是数据传输最重要的两种协议，但是它们的性能在传输大量的数据时并不令人满意。一些增强 TCP 功能的方法的目标是提高链路吞吐率，并对长短不一的混合 TCP 流提供可预测的小延迟。例如，DCTCP 利用显示拥塞通知对端主机提供多比特反馈。UDP 协议适用于传输大量数据，但是缺乏拥塞控制。因此高带宽的 UDP 应用必须自己实现拥塞控制机制，这是一项困难的任务并且会导致风险。

（二）数据预处理

数据源具有多样性，数据集因干扰、冗余和一致性因素的影响而具有不同的质量。从需求的角度来看，一些数据分析工具和应用对数据质量有着严格的要求。因此，在大数据系统中需要使用数据预处理技术来提高数据的质量。

主要的数据预处理技术包括数据整合、数据清洗、冗余消除、数据归约等。

1. 数据整合

数据整合是指在逻辑上和物理上把来自不同数据源的异构数据进行连接合并，为用户提供一个统一的数据视图。这些不同来源的异构数据可能存在命名和格式不统一、数据重复、数据类型不一致等问题，因此，需要根据一定的规则将这些数据进行必要的处理和格式转换，然后进行连接合并，形成统一的数据视图。

2. 数据清洗

数据清洗（cleaning）是指在数据集中发现不准确、不完整或不合理的数据，并对这些数据进行修补或删除以提高数据质量。一个通用的数据清洗过程由 5 个步骤构成：定义错误类型，搜索并标识错误实例，改正错误，文档记录错误实例和错误类型，修改数据输入程序以减少未来的错误。

此外，格式检查、完整性检查、合理性检查和极限检查也在数据清洗过程中完成。数据清洗对保持数据的一致和更新起着重要作用，因此广泛应用于银行、保险、零售、电信和交通等多个领域。在电子商务领域，尽管大多数数据通过电子方式收集，但仍存在数据质量问题。影响数据质量的因素包括技术、业务和管理三个方面，技术因素涉及数据来源、数据采集、数据传输和数据装载等方面，业务因素涉及业务不清晰、输入不规范、数据造假等方面，管理因素涉及人员素质、管理机制、数据规范、流程制度等方面。

数据清洗对随后的数据分析非常重要，因为它能提高数据分析的准确性。但是数据清洗依赖复杂的关系模型，这会带来额外的计算和延迟开销，因此，必须在数据清洗模型的复杂性和分析结果的准确性之间进行平衡。

3. 冗余消除

数据冗余是指数据的重复或过剩，这是许多数据集的常见问题。数据冗余无疑会增加传输开销，浪费存储空间，导致数据不一致，降低可靠性。因此，许多研究提出了数据冗余减少机制，如冗余检测和数据压缩。

由广泛部署的摄像头收集的图像和视频数据存在大量的数据冗余。在视频监控数据中，大量的图像和视频数据存在着时间、空间和统计上的冗余。视频压缩技术被用于减少视频数据的冗余，许多重要的标准（如 MPEG－2，MPEG－4，H.263，H.264/AVC）已被应用以减少存储和传输的负担。

对于普遍的数据传输和存储，数据去重技术是专用的数据压缩技术，用于消除重复数据的副本。数据去重技术能够显著地减少存储空间的占用，对大数据存储系统具有非常重要的作用。

4. 数据归约

数据整合与清洗无法改变数据集的规模，依然需要通过技术手段降低数据规模，这就是数据归约。数据归约采用编码方案，通过小波变换或主成分分析来有效地压缩原始数据，或者通过特征提取技术进行属性子集的选择或重造。

除了前文提到的数据预处理方法，还有一些对特定数据对象（这些数据对象通常具有高维特征矢量）进行预处理的技术，如特征提取技术，在多媒体搜索和域名系统分析中起着重要作用。数据变形技术则通常用于处理分布式数据源产生的异构数据，对商业数据的处理非常有用。然而，没有一个统一的数据预处理过程和单一的技术能够用于多样化的数据集，必须考虑数据集的特性、需要解决的问题、性能需求和其他因素来选择合适的数据预处理方案。

知识拓展

数据存储

数据存储是指数据以某种格式记录在计算机内部或外部存储介质上。它包括两部分，即存储格式与存储介质。

1. 存储格式

文件：文字文件，压缩文件，图形图像，动画，音频、视频文件等。

数据库：关系型数据库，非关系型数据库。

2. 存储介质

磁盘和磁带都是常用的存储介质。数据存储组织方式因存储介质而异。在磁带上数据仅采用顺序存取方式；在磁盘上则可按使用要求采用顺序存取或直接存取方式。数据存储方式与数据文件组织密切相关，其关键在于建立记录的逻辑与物理顺序间的对应关系，确定存储地址，以提高数据存取速度。

任务二 网络爬虫

任务描述

网络爬虫是一种按照一定规则自动爬取互联网信息的程序或脚本，它针对既定的爬取目标有选择地访问网页及相关链接，获取需要的数据资源。本任务我们学习网络爬虫的相关知识。

任务目标

1. 能掌握网络爬虫的相关知识。
2. 会用 requests 库与抓包工具的结合实现一个 App 页面内容的爬取。
3. 培养获取信息并利用信息的能力。
4. 守正创新，培养科技强国的信念。

任务实施

一、网络爬虫的概念与应用现状

（一）网络爬虫的概念

网络爬虫可以理解为在网络上爬行的一只蜘蛛。若将互联网比作一张大网，则爬虫便是在这张网上爬来爬去的蜘蛛，如果遇到资源，它就会爬取下来。

比如，网络爬虫在爬取一个网页时发现了一条通往其他网页的道路（即指向网页的超链接），那么它就可以爬到另一个网页以获取数据。这样，连在一起的整个大网对网络爬虫来说就触手可及了。本书重点介绍的 Python 爬虫就是利用 Python 语言实现的网络爬虫。

例如，想象一下平时到天猫商城购物（PC 端）的步骤，可能是打开浏览器→搜索天猫商城→单击链接进入天猫商城→选择所需商品类目（站内搜索）→浏览商品（价格、详情参数、评论等）→单击链接→进入下一个商品页面……周而复始。除了天猫商城之外，京东商城、苏宁易购等的相关操作也类似上述步骤，当然其中的搜索商品也是爬虫的应用之一。简单地讲，网络爬虫是类似又区别于上述场景的一种程序。以上场景属于传统爬虫：传统爬虫从一个或若干个初始网页的统一资源定位符（Uniform Resource Locator，URL）开始获得初始网页上的 URL，在爬取网页的过程中不断从当前页面上抽取新的 URL 放入队列，直到满足停止条件。

通过网络爬虫的学习，同学们可以设计并实现一个小型的搜索引擎，这可以通过编写自己的爬虫来实现。当然，这个爬虫在性能或者算法上肯定比不上主流的搜索引擎，但是其个性化程度会非常高，并且有利于我们更深层次地理解搜索引擎内部的工作原理。

（二）网络爬虫的应用现状

网络爬虫的应用源于 20 世纪 90 年代的 Google 等搜索引擎，爬虫用于抓取互联网上的 Web 页面，再由搜索引擎进行索引和存储，从而为网民提供信息检索服务。在系

统架构上，网络爬虫位于搜索引擎的后台，并未直接与网民接触，因此在较长的时间内并未被广大开发人员所关注，相应的技术研究也很有限。

2004 年以前，我国对网络爬虫技术和应用的关注度几乎为 0，但自 2005 年以来人们对网络爬虫技术的关注度快速上升。进一步分析发现，对网络爬虫技术及应用的关注度排名在前面的领域依次是计算机软件及计算机应用、互联网技术与自动化技术、新闻与传媒、贸易经济、图书情报与数字图书馆、企业经济、自然地理学和测绘学、金融投资，其中超过 90% 的关注度主要集中在前两者，它们侧重于爬虫技术研究，紧接在后面的是主要的网络应用领域，可以看出爬虫技术的应用领域很广泛。

爬虫是一个实践性很强的技术活，因此网络爬虫技术关注度的变化趋势也从另一个角度反映了互联网上运行的爬虫数量的增长速度。除了为数不多的主流互联网搜索引擎爬虫外，互联网上运行的爬虫主要来自个人、中小型企业和单位。

爬虫应用的迅速普及得益于大量的网络爬虫开源包或底层技术开源包的出现，这些开源包使得开发一个具体应用的网络爬虫采集系统变得容易很多。但是，也正是由于这个原因，高度封装的开源包使得很少有人愿意深入了解其中的关键技术，导致这种途径生成的爬虫质量、性能和友好程度都受到很大影响。甚至网络爬虫也因此被认为是一个不太“优雅”的行业，当然这种看法并不正确，不能被低质量的个人或小型爬虫迷惑而看不清行业现状。相反，我们应当深入分析导致这种问题的技术和非技术因素，制定更为完善的爬虫大数据采集规范或要求。

目前，低质量的个人、小型爬虫存在的主要问题可以归结为以下 3 个方面。

（1）不遵守 Robots 协议，连接一个 Web 服务器之后不检测虚拟根目录下是否存在 robots.txt 文件，也不管文件里面关于页面访问许可列表的规定。由于这个协议是一个行业规范，忽视或不遵守这个协议也就意味着行业的发展会进入不良状态。

（2）爬行策略没有优化，一般开源系统实现了宽度优先或深度优先的策略，但是并没有对 Web 页面的具体特征做优化，例如 Portal 类型页面的超链接非常多，这些链接如果直接进入爬行任务，就很容易对 Web 服务器造成拒绝服务攻击。

（3）许多爬虫实现了多线程、分布式的架构，这个看似好的软件架构技术对于网络爬虫来说可能只是“一厢情愿”。客户端架构设计得再好，爬行策略、增量模式等问题没有解决好，其效果就相当于制造了很多小爬虫在服务器上同时运行。这种情况最终导致两败俱伤的结局，Web 服务器需要投入大量的人力、物力和资金进行爬虫检测和阻断，对于爬虫也一样，因此最终对 Web 服务器和采集数据的爬虫都不利。

二、网络爬虫的结构与组成

（一）网络爬虫的结构

网络爬虫的基本结构及工作流程如图 1–3 所示。

（1）首先选取一部分精心挑选的种子 URL。

（2）将这些 URL 放入待爬取的 URL 队列。

（3）从待爬取的 URL 队列中取出待爬取的 URL，解析 DNS（Domain Name System，

域名解析系统），并得到主机的网络协议地址（Internet Protocol Address，IP 地址），将 URL 对应的网页下载下来并存储到已下载的网页库中，然后将这些 URL 放入已爬取的 URL 队列。

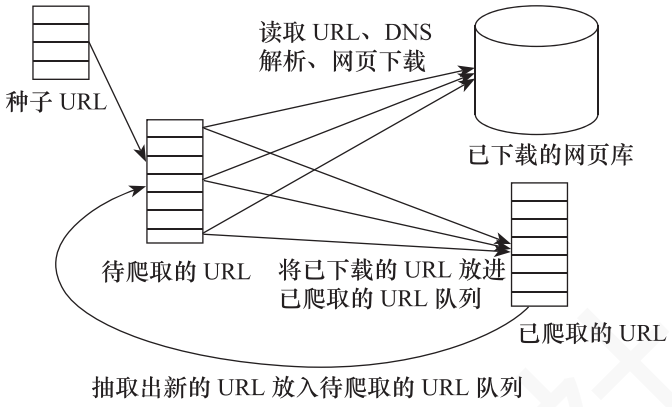


图 1-3　通用的网络爬虫框架

（4）分析已爬取的 URL 队列中的 URL 和其中的其他 URL，并将 URL 放入待爬取的 URL 队列，从而进入下一个循环。

（二）网络爬虫的组成

网络爬虫由控制节点、爬虫节点、资源库构成。网络爬虫的控制节点和爬虫节点的结构关系如图 1-4 所示。

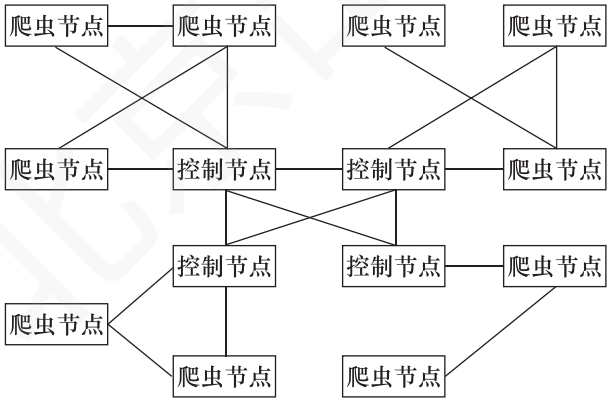


图 1-4　网络爬虫的控制节点和爬虫节点的结构关系

从图 1-4 可以看出，网络爬虫可以有多个控制节点，每个控制节点下又可以有多个爬虫节点，控制节点之间可以互相通信，控制节点和其下的各爬虫节点也可以互相通信，属于同一个控制节点下的各爬虫节点亦可以互相通信。

控制节点也称爬虫的中央控制器，主要负责为 URL 地址分配线程，并调用爬虫节点进行具体的爬行。

爬虫节点会按照相关的算法对网页进行具体的爬行，主要包括下载网页及对网页的文本进行处理，并在爬行后将对应的爬行结果存储到对应的资源库中。

三、网络爬虫的类型

网络爬虫的类型可以分为通用网络爬虫、聚焦网络爬虫、增量式网络爬虫和深层网络爬虫。

（一）通用网络爬虫

通用网络爬虫又称全网爬虫（Scalable Web Crawler，SWC），爬行对象从一些种子URL扩充到整个Web，该架构主要为门户网站搜索引擎和大型Web服务提供商采集数据。为提高工作效率，通用网络爬虫会采取一定的爬行策略，常用的爬行策略有深度优先策略和广度优先策略。

1. 深度优先策略

深度优先策略按照深度由低到高的顺序依次访问下一级网页链接，直到不能再深入为止。爬虫在完成一个爬行分支后会返回上一个链接节点，再进一步搜索其他链接。当所有链接遍历完成后，爬行任务结束。这种策略比较适合垂直搜索或站内搜索，但在爬行页面内容层次较深的站点时会造成巨大的资源浪费。

2. 广度优先策略

广度优先策略按照网页内容目录层次的深浅爬行页面，处于较浅目录层次的页面会首先被爬行。当同一层次中的页面爬行完毕后，爬虫再深入下一层继续爬行。这种策略能够有效控制页面的爬行深度，避免在遇到一个无穷深层分支时无法结束爬行的问题，实现方便，无须存储大量中间节点；其不足之处在于需要较长时间才能爬行到目录层次较深的页面。

（二）聚焦网络爬虫

聚焦网络爬虫（Focused Crawler，FC）又称主题网络爬虫（Topical Crawler，TC），是指选择性地爬行那些与预先定义好的主题相关的页面的网络爬虫，常用的策略有基于内容评价的爬行策略、基于链接结构评价的爬行策略、基于增强学习的爬行策略和基于语境图的爬行策略。

1. 基于内容评价的爬行策略

为了将文本相似度的计算方法引入网络爬虫，人们提出了Fish－Search算法，它将用户输入的查询词作为主题，包含查询词的页面被视为与主题相关。该算法的局限性在于无法评价页面与主题相关度的高低。后来人们对Fish－Search算法进行了改进，提出了Shark－Search算法，它利用空间向量模型计算页面与主题的相关度大小。

2. 基于链接结构评价的爬行策略

Web页面作为一种半结构化文档，包含很多结构信息，可以用来评价链接的重要性。PageRank算法（又称网页排名）最初用于在搜索引擎信息检索中对查询结果进行排序，也可用于评价链接的重要性，具体做法是每次选择PageRank值较大的页面中的链接进行访问。另一个利用Web结构评价链接价值的算法是超文本敏感标题搜索（Hyperlink－Induced Topic Search，HITS）算法，它通过计算每个已访问页面的Authority

权重和 Hub 权重来决定链接的访问顺序。

3. 基于增强学习的爬行策略

Rennie 和 McCallum 将增强学习引入聚焦爬虫，利用贝叶斯分类器，根据整个网页文本和链接文本对超链接进行分类，对每个链接计算出重要性，从而决定链接的访问顺序。

4. 基于语境图的爬行策略

Diligenti 等人提出了一种通过建立语境图（Context Graphs）学习网页之间的相关度并训练机器进行学习的系统，通过该系统可以计算当前页面到相关 Web 页面的距离，距离越近的页面中的链接会被优先访问。

（三）增量式网络爬虫

增量式网络爬虫（Incremental Web Crawler，IWC）是指对已下载的网页采取增量式更新和只爬行新产生或者已经发生变化的网页的爬虫，它能够在一定程度上保证爬行的页面是尽可能新的页面。

增量式爬虫有两个目标：保持本地页面集中存储的页面为最新页面和提高本地页面集中存储页面的质量。为实现第一个目标，增量式爬虫需要通过重新访问网页来更新本地页面集中的页面内容。常用的方法如下。

（1）统一更新法：爬虫以相同的频率访问所有网页，不考虑网页的改变频率。

（2）个体更新法：爬虫根据个体网页的改变频率重新访问各页面。

（3）基于分类的更新法：爬虫根据网页的改变频率将其分为更新较快的网页子集和更新较慢的网页子集，然后以不同的频率访问这两类网页。

为实现第二个目标，增量式爬虫需要对网页的重要性进行排序，常用的策略有广度优先策略和 PageRank 优先策略。

（四）深层网络爬虫

深层网络爬虫将 Web 页面按存在的方式分为表层网页（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，即以超链接可以到达的静态网页为主构成的 Web 页面。深层网页是指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。深层网络爬虫体系结构包含 6 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 2 个爬虫内部数据结构（URL 列表、LVS 表）。其中，LVS（Label Value Set）表示标签 / 数值的集合，用来表示填充表单的数据源。

爬取过程中最重要的部分就是表单填写，包含以下两种类型。

（1）基于领域知识的表单填写。

（2）基于网页结构分析的表单填写。

实际的网络爬虫系统通常是由几种爬虫技术相结合实现的。

网络爬虫是采集数据的一门技术，它可以帮助人们自动进行信息的获取与筛选。从技术手段来讲，网络爬虫有多种实现技术，如 PHP、Java、Python 等，用 Python 也会有很多不同的技术方案（Urllib、Requests、Scrapy、Selenium 等），每种技术各有各的特点，同学们只需要掌握一种技术，其他的便能够融会贯通。

四、网络爬虫的相关技术

（一）相关协议与规范

网络爬虫是一种客户端技术，它不能离开服务端独立工作，而服务端是由众多分布在互联网上的 Web 服务器组成的。在这样的环境下爬虫要从不同的配置、不同 Web 软件的服务器上采集页面信息，就需要按照一定的协议或规范来完成交互过程。在爬虫技术实现时需要遵守这些协议或规范。具体来讲，主要有如下这些协议与规范。

1. TCP 协议

TCP 协议是网络爬虫的底层协议，当爬虫与 Web 服务器建立连接、传输数据时都是以该协议为基础。在技术实现上具体表现为 Socket 编程技术。各种语言都提供了对此的支持，如 Java 提供 InetAddress 类，可以完成对域名与 IP 之间的正向、逆向解析。在 Python 中则有 dnspython 这个 DNS 工具包，利用其查询功能可以实现 DNS 的服务监控及解析结果的校验。不管哪种语言或开发平台，DNS 的解析一般都是调用系统自带的 API，通常是 Socket 的 getaddrinfo（）函数。

2. HTTP 协议

HTTP 协议是一种应用层协议，用于超文本传输。它规定了在 TCP 连接上向 Web 服务器请求页面以及服务器向爬虫响应页面数据的方式和数据格式。爬虫实现时，对 HTTP 协议的依赖比较大，目前 Web 服务器使用的 HTTP 协议版本主要是 HTTP 1.0 和 HTTP 1.1，而最新的版本是 HTTP 2.0。这些协议在功能上有一定差异，但也有很多共同的地方。在设计爬虫程序时需要充分了解 HTTP 协议。

3. Robots 协议

Robots 协议也称为爬虫协议，其全称是“网络爬虫排除协议”（Robots Exclusion Protocol）。该协议指明了哪些页面可以抓取，哪些页面不能抓取，以及抓取动作的时间、延时、频次限定等。该协议最早是针对搜索引擎爬虫，目前在各种爬虫中都可适用。Robots 协议只代表了一种契约，并不是一种需要强制实施的协议。爬虫遵守这一规则，能够保证互联网数据采集的规范化，有利于行业的健康发展。

4. Cookie 规范

Cookie 是指某些网站为了辨别用户身份、进行 Session 跟踪而存储在用户本地设备

上的数据。通过 Cookie 可以将用户在服务端的相关信息保存在本地，这些信息通常是用户名、口令、地区标识等，这些信息会由浏览器自动读出，并通过 HTTP 协议发送到服务端。在 RFC 6265 规范中具体规定了 Cookie 的数据含义、格式和使用方法。

5. 网页编码规范

网页编码是指对网页中的字符采用的编码方式。由于一个网页可能被来自世界各地的访客访问，而每个国家的语言并不完全相同，所以为了使网页内容能正常显示在访客的浏览器上，需要有一套共同的约定来表明页面中字符的提取识别方法。目前常见的网页字符编码主要有 unicode、utf-8、gbk、gb2312 等，其中 utf-8 为国际化编码，在各国各地区的网站中都很常见，是最通用的字符编码。爬虫在解析页面内容时就需要识别页面的编码方式。

6. HTML 语言规范

HTML（Hyper Text Markup Language，超文本标记语言）是一种用来描述网页的语言，它规定了页面的版式、字体、超链接、表格，甚至音乐、视频、程序等非文字元素的表示方法。对爬虫采集页面的解析、对表格数据的提取、对正文的提取等都需要根据 HTML 定义的各种标签才能正确完成。

（二）Web 信息提取技术

Web 信息提取技术

对于网络爬虫采集页面数据而言，最终的目标是获得页面中的内容，因此如何从 HTML 编码的内容提取所需要的信息是爬虫采集 Web 数据需要解决的问题。此外，由于爬虫是依赖于超链接来获得更多的爬行页面，所以从 Web 页面中提取超链接也是 Web 信息提取的技术问题。

总的来讲，Web 信息提取包含两大部分，即 Web 页面中的超链接提取和 Web 内容提取。对于 Web 页面中的超链接提取而言，超链接在 Web 页面中具有相对比较有限的标签特征，因此通常可以使用简单的正则表达式之类的方法来提取。对于页面中正文内容的提取则要复杂一些。由于不同页面的正文位置并不相同，并且网站也会经常改版，所以为了爬虫解析 Web 页面的程序能够具备一定的灵活性和适应性，需要引入一定的技术手段来减轻这种变化所需要的程序维护工作。其常用的方法是将 Web 页面转换成为一棵树，然后按照一定的规则从树中获得所需要的信息。

目前，在 Python 中已经有很多种开源库可以用来实现基于树结构的信息提取，并且有灵活的策略可以配置。这些开源库主要有 html. parser、lxml、html5lib、BeautifulSoup 以及 PyQuery 等。这些库各有各的优缺点，开发人员在实际应用中可以选择合适的方法。

一些高级的方法试图使爬虫提取 Web 信息有更好的适应能力，一种途径是引入统计思想，对页面中正文部分的各种特征进行统计，在大量样本特征计算的基础上设置合适的特征值范围，从而为自动提取提供依据。

（三）典型应用中的数据获取技术

网络爬虫有多种不同的采集需求，Deep Web 和主题获取是其中的两种典型代表，

在实际中也会经常用到。其中所涉及的技术与普通爬虫并不一样，因此开发人员通常需要全面掌握。

能够进行主题获取的爬虫被称为主题爬虫，在技术手段上，其核心在于主题。围绕主题的定义方法、主题相似度计算等关键问题，有一系列来自文本内容分析的技术可以使用，主要有文本预处理技术、主题表示、主题建模等。文本预处理技术包括词汇的切分、停用词过滤等，而主题建模则可以利用各种主题模型，如 PLSA、LDA、基于向量空间的主题表示模型以及简单的布尔模型等。

Deep Web 采集的目标是获得存储在后台数据库中的数据，属于一种深度数据获取，而普通爬虫通常是一种面向表面的数据采集。既然是深度数据获取，就需要对数据的采集接口有一定的处理能力，同时需要具备一定的输入识别和自动填写能力，因此需要一定的知识库来支持。

（四）网络爬虫的软件技术

除了技术架构中所列出来的技术外，对网络爬虫而言另一个重要的问题是这些技术如何进行协调合作，共同完成互联网大数据的采集。这是通过网络爬虫的软件技术来保证的，在技术实现时通常有多种不同的选择。

1. 多线程技术

从网络爬虫的技术体系看，对于某个页面的采集，3 个层次上的功能执行具有先后顺序，即必须先建立网络连接，再进行 HTTP 协议数据的发送和接收处理，最后根据爬虫采集需求对接收到的页面数据进行解析和内容提取。如果是主题爬虫，还需要进行一些内容分析。

爬虫通常并不针对某个页面，而是根据超链接爬取多个页面。这些页面的爬取过程之间相互独立，因此在实现时可以使用多线程技术。通常的做法是设置若干线程分别进行页面内容提取、URL 处理、HTTP 命令数据包构建、响应数据的接收以及建立网络连接等。不同线程之间可以通过文件、共享内存进行数据交换。

2. 单机系统

如果需要爬取的页面数量不多，在爬虫系统的技术实现上可以部署在一台机器上，即单机系统模式。在这个模式下，线程的设置要考虑到机器的配置和网络带宽。如果配置高，则线程数量可以多一些；如果网络带宽大，则处理网络连接的线程可以多一些。具体线程数量需要在实际环境下进行调整。

3. 分布式系统

如果需要爬取的页面数量很多，以至于爬虫很难在用户预期的时间内完成页面数据的采集，在这种情况下就需要进行分布式处理。在分布式爬虫系统设计中，一般将爬行任务（即 URL 列表）分配给若干个不同的计算节点，并设置统一的协调中心来管理整个分布式系统所要爬行的 URL 列表。在分布式系统中，每个节点在处理爬行任务时仍可以采用多线程结构。

尽管爬虫在软件技术方面有多种不同的选择，但爬虫只是一个客户端程序，为了有

效提高整个爬虫系统采集数据的性能，显然不能忽略服务器端的承受和响应能力。对于爬虫系统而言，它是根据所要爬行的 URL 集合来执行任务的。如果在短时间内，爬虫端多个线程或分布节点同时连接到同一个服务器进行页面采集，显然这些大量的连接请求会在服务器端产生较大的资源占用，从而影响服务器的正常运行，最终导致爬虫系统采集数据的效率降低。因此，在多线程、分布式爬虫设计时应当进行合理的爬行任务分配，即设计合理的爬行策略，以避免这种情况出现。

五、网络爬虫数据采集与挖掘的合规性

随着网络爬虫技术应用的普及，网络爬虫的应用场景越来越多，但是不合理使用网络爬虫技术进行大数据采集的案例也不断出现，甚至导致了相应的法律问题。因此，网络爬虫能以什么方式爬取什么数据这个问题是值得考虑的。

（一）数据爬取权限

从数据爬取权限方面来看，爬虫爬取的数据是否具有访问权限是其边界之一。访问权限可以从数据是否公开、页面是否许可来判断。爬虫对公开的数据当然具备爬取权限，公开或不公开的判断依据为是否需要以一定用户身份登录后才能看到数据，并且以其他用户身份登录后是看不到数据的。在各类不公开数据的采集中，容易引起纠纷的是用户个人信息，包括个人身份信息、行踪轨迹、联系方式等。在采集这类数据前，爬虫应当获得用户授权。

未公开的网络数据，爬虫程序无权获取，否则可能会被认定为非法获取计算机信息系统数据罪。在《中华人民共和国刑法》第二百八十五条提到非法获取计算机信息系统数据罪，是指侵入国家事务、国防建设、尖端科学技术领域以外的计算机信息系统或者采用其他技术手段，获取该计算机信息系统中存储、处理或者传输的数据。这里“侵入”是指行为人采用破解密码、盗取密码、强行突破安全工具等方法，在没有得到许可时违背计算机信息系统控制人或所有人意愿进入其无权进入的计算机信息系统中。典型的途径是破解 App 的加密算法或网络交互协议、调用规则和参数，从而爬虫突破权限许可获取数据。

爬取权限的另一个界定方法是 Robots 协议，如果网站有设置 robots.txt 文件，则爬虫应当依据该文件决定某个特定的 URL 是否许可。

（二）爬虫的访问方式

爬虫的访问方式是指爬虫访问服务器的方式，其边界为爬虫是否对服务器的正常运行造成影响。如果网络爬虫在短时间内频繁访问 Web 服务器，通常是采用分布式、并行爬取等技术，从而导致服务器不能正常运行，其客户访问变得很慢甚至无法响应。如果突破这个边界，可能会涉及破坏计算机信息系统罪，目前也有一些爬虫爬取案件被法院按这种类型处理。

与访问方式有关的另一个边界仍然是 Robots 协议，在该协议中定义了爬取延时、爬取时间段等参数，如果爬虫没有遵守这些约定，则可能导致服务器不能正常运行。不过，据观察，许多网站并没有充分运用 Robots 协议来定义这些参数。

（三）数据量与数据的使用

数据使用边界是指爬取的数据是否用于商业用途、是否涉及版权限定。例如，百度公司通过爬虫技术从大众点评网等网站获取信息，并将爬取的信息直接提供给网络用户（展示），最终被上海知识产权法院认定为不正当竞争行为。虽然百度公司的搜索引擎爬取涉案信息并不违反 Robots 协议，但是将大量数据用于商业用途或展示传播，很可能会涉及不正当竞争，属于利益冲突。此外，根据个人信息安全规范，涉及个人信息的数据不应该存储在本地或进行融合挖掘。

总的来看，互联网公开资源的爬取并不违法，网络爬虫作为互联网大数据采集的技术手段，本身具有中立性，而爬取没有权限、没有授权的数据，对服务器正常运行产生影响，以及爬取后的数据用于商业用途、未经授权公开展示，则是突破了爬虫大数据采集的边界。

与爬虫大数据采集相关的规范和法律条款主要出现在《中华人民共和国网络安全法》《中华人民共和国计算机信息系统安全保护条例》《个人信息安全规范》《中华人民共和国数据安全法》和《中华人民共和国反不正当竞争法》中。在设计爬虫大数据采集挖掘系统之前建议阅读这些规范和法律条款，设计方案时要对合规性和采集性进行适当的平衡，不能为了提高采集性而忽视合规性。

六、Scrapy 爬虫

Scrapy 是一个为了爬取网站数据、提取结构化数据而编写的应用框架，可以应用在包括数据挖掘、信息处理或存储历史数据等一系列的程序中，其最初是为了页面爬取（更确切来说是网络爬取）而设计的。

Scrapy 是一套用 Python 编写的异步爬虫框架，基于 Twisted 实现，运行于 Linux、Windows、Mac OS 等多种环境，具有速度快、扩展性强、使用简便等特点。即便是新手也能迅速学会使用 Scrapy 编写需要的爬虫程序。Scrapy 可以在本地运行，也可以部署到云端，从而实现真正的生产级数据采集系统。

（一）Scrapy 框架

Scrapy 框架是一套比较成熟的 Python 爬虫框架，是使用 Python 开发的快速、高层次的信息爬取框架，可以高效地爬取 Web 页面并提取出结构化数据。Scrapy 用途广泛，可以用于数据挖掘、监测和自动化测试。Scrapy 吸引人的地方在于它是一个框架，任何人都可以根据需求对它进行修改。Scrapy 的整体构架大致如图 1－5 所示。

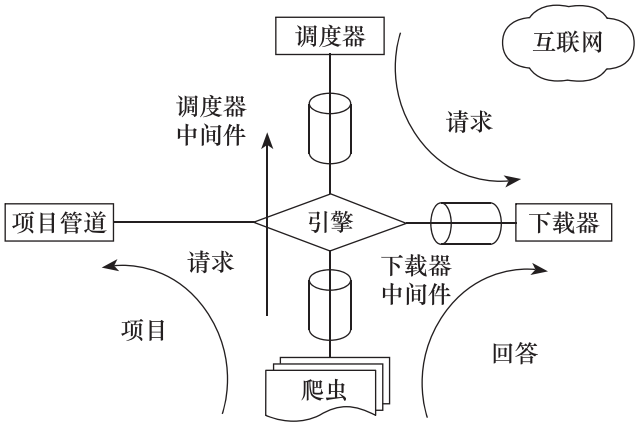


图 1-5　Scrapy 的整体构架

（二）Scrapy 的常用组件

Scrapy 是一个爬虫框架，由很多组件构成，具体如下。

1. 引擎

引擎（Engine）用来处理整个系统的数据流，触发事务（框架核心），负责与各种组件交流。

2. 调度器

调度器（Scheduler）用来接收引擎发来的 Request（请求），压入队列并在引擎再次请求时返回，可以将其想象成一个 URL（爬取网页的网址或者说链接）的优先队列，由它决定下一个要爬取的网址是什么，同时去除重复的网址。

3. 下载器

下载器（Downloader）用于下载搜索引擎发送的所有请求，并将网页内容返回给爬虫。下载器建立在 Twisted 这个高效的异步模型之上。

4. 爬虫

爬虫（Spider）负责处理所有应答包（Response），从中分析和提取数据，获取 Item（项目）字段需要的数据或链接，并将需要跟进的 URL 提交给引擎，再次进入 Scheduler（调度器）。

5. 项目管道

项目管道（Item Pipeline）用来保存数据，负责处理爬虫中获取到的 Item 并进行处理，如去重、持久化存储（存数据库，写入文件）。

6. 下载器中间件

下载器中间件（Downloader Middlewares）是位于引擎和下载器之间的框架，主要用于处理引擎与下载器之间的请求及响应，类似于自定义扩展下载功能的组件。

7. 爬虫中间件

爬虫中间件（Spider Middlewares）是位于引擎和爬虫之间的框架，主要用于处理爬

虫的响应输入和请求输出。

8. 调度器中间件

调度器中间件（Scheduler Middlewares）是位于引擎和调度器之间的中间件，用于处理从引擎发送到调度器的请求和响应，可以自定义扩展和操作搜索引擎与爬虫中间“通信”的功能组件（如进入爬虫的请求和从爬虫出去的请求）。

（三）Scrapy 工作流

Scrapy 工作流也称运行流程或数据处理流程，整个数据处理流程由 Scrapy 引擎控制，其主要运行步骤如下。

（1）Scrapy 引擎（Scrapy Engine）从调度器中取出一个链接，用于接下来的爬取。

（2）Scrapy 引擎把 URL 封装成一个请求并传给下载器。

（3）下载器把资源下载下来，并封装成应答包。

（4）爬虫解析应答包。

（5）若解析出项目，则交给项目管道（Item Pipeline）进行进一步的处理。

（6）若解析出链接，则把 URL 交给调度器等待爬取。

（四）其他 Python 框架

Scrapy 只是 Python 的一个主流框架，除了 Scrapy 外，还有其他的主流框架，下面简要介绍几种主流框架及其特点。

1. Crawley

Crawley 可高速爬取网站的内容，支持关系和非关系数据库，数据可以导出为 JavaScript（简称 JS）对象简谱（JavaScript Object Notation，JSON）、可扩展标记语言（Extensible Markup Language，XML）等格式。

2. Portia

Portia 用来可视化爬取网页内容。

3. Newspaper

Newspaper 用来爬取新闻、文章及内容分析。

4. Python-goose

Python－goose 是用 Java 语言编写的爬取网页中文章的工具。

5. Beautiful Soup

Beautiful Soup 整合了一些常用的爬虫需求，缺点是不能加载 JS。

6. Mechanize

Mechanize 的优点是可以加载 JS，缺点是文档严重缺失。不过通过官方示例及人工尝试的方法，Mechanize 还是勉强能用的。

7. Selenium

Selenium 是一个调用浏览器的服务器库，通过该库可以直接调用浏览器完成某些操作，如输入验证码。

8. Cola

Cola 是一个分布式爬虫框架。

知识拓展

爬虫技术评价方法

爬虫技术是一种典型的 Web 页面数据采集方法，得到了许多技术人员的关注，因此目前不断有新的爬虫技术或开源框架被提出来。在这种情况下需要有一套比较完整的爬虫技术评价方法，以便于进行比较、权衡和选择。归纳起来，网络爬虫技术的评价方法可以从以下 10 个方面进行，这 10 个方面可以分为友好爬虫技术（1、2）、页面采集技术（3、4、5）、内容处理技术（6、7）、爬虫软件技术（8、9、10）4 个方面。

1. 是否遵守 Robots 协议

在 Web 页面抓取的过程中，是否进行了 Robots 许可公告的判断，是否根据许可规范来确定爬虫的抓取权限。

2. 友好爬虫请求技术

友好爬虫以不对 Web 服务器造成拒绝服务攻击为底线，因此友好爬虫应当具备请求间隔可调整、符合 Web 服务器关于访问高峰期的规定，同时应当根据服务器返回的状态码及时调整自己的请求强度。

3. 高效采集技术

高效是指在一定的时间和网络带宽限定下爬虫采集到尽可能多的 Web 页面。其中所涉及的爬虫核心技术较多，包括站内页面的遍历策略、站外页面的遍历策略、URL 去重技术等。对于 Deep Web 来说，还涉及如何降低查询次数的技术问题。

4. 对增量式采集的支持

在每次运行时，爬虫是否能够判断哪些页面内容已经更新，并采集自上次采集以来新出现的内容，此即为增量式采集技术。

5. 对动态页面的支持

动态页面的实现可以通过 URL 传递参数、通过 Cookie 传递参数以及使用 Ajax 等技术来实现，不同的爬虫技术对这些技术的支持程度有所不同。

6. 页面编码与语言处理能力

普通型的网络爬虫根据超链接在 Web 空间上跳转，很可能要面对多种不同页面、不同语言的 Web 页面，好的爬虫应当能够处理这些差异可能造成的存储信息乱码问题。

7. 主题相关度评估

对于主题爬虫而言，要衡量采集到的页面与事先设定的主题的相关度，可以进一步从主题信息的召回率和准确率两个指标来衡量。

8. 对分布式架构的支持

在面对海量 Web 信息的采集任务时，通常需要爬虫具备分布式架构，以协调多台

计算机高效完成采集任务。

9. 可配置线程技术

爬虫需要完成 Web 服务器连接建立、URL 命令发送、Web 页面内容采集、URL 过滤以及爬行策略管理等任务，这些任务可以按照一定方式同步进行，从而提升采集效率。可以根据计算机的配置情况来设定线程数量是一个必要的技术。

10. 容错能力

在一般情况下，爬虫采集时需要面对 Web 服务器、通信网络等多方面的异常，健壮的爬虫应当具有一定的容错能力，以避免各个环节上的错误而导致爬虫系统崩溃。

实训 爬取手机端数据

一、实训目标

使用 requests 库与抓包工具（拦截查看网络数据包内容的软件）的结合实现一个 App 页面内容的爬取。能够通过 Fiddler 抓包工具的配置及使用获取 App 数据内容及相关信息，之后使用 requests 库的相关方法通过链接地址实现对 App 内数据的爬取。爬取思路如下。

（1）安装 Fiddler 抓包工具。

（2）使用 Fiddler 抓包工具进行网站分析。

二、实训操作

（一）下载抓包工具

进入 Fiddler 抓包工具官网，根据相关提示信息完成内容填写后，单击“Download For Windows”按钮后，即可下载 Fiddler 工具，如图 1-6 所示。

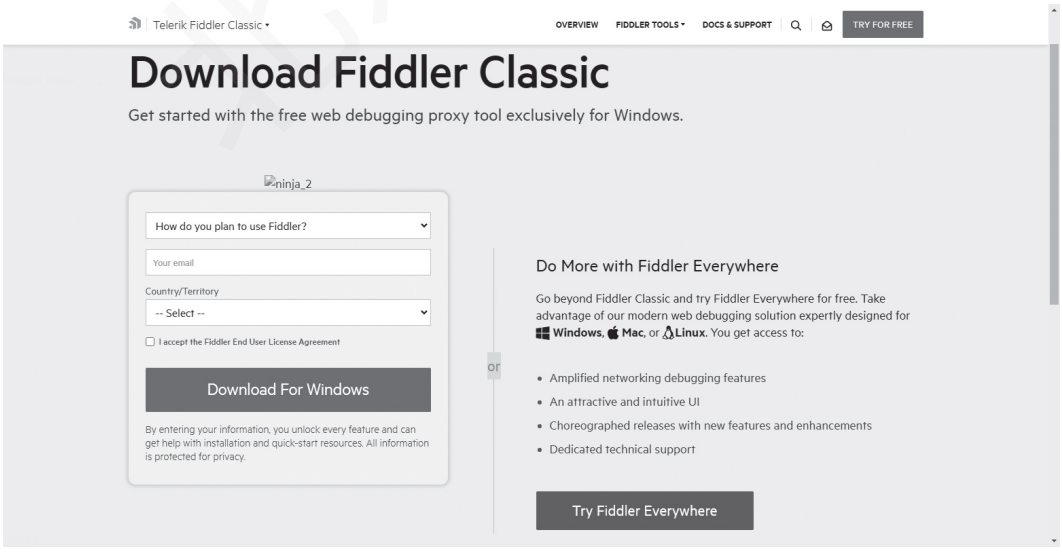


图 1-6 下载抓包工具

（二）安装 Fiddler

双击下载好的软件安装包，单击“Install”按钮即可安装 Fiddler 工具，如图 1-7 所示。

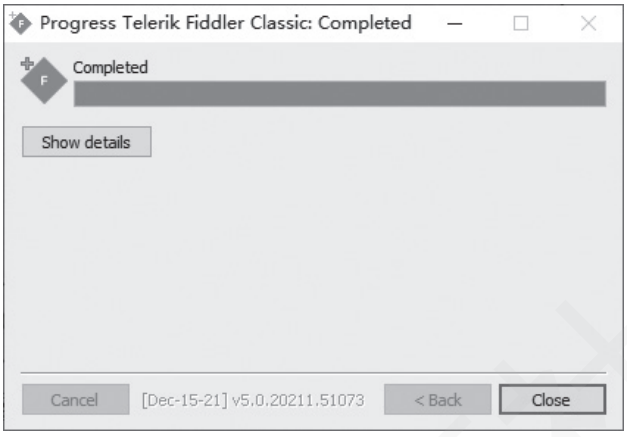


图 1-7　安装 Fiddler 工具

（三）配置 Fiddler 工具

打开安装完成的 Fiddler 软件，如图 1-8 所示。

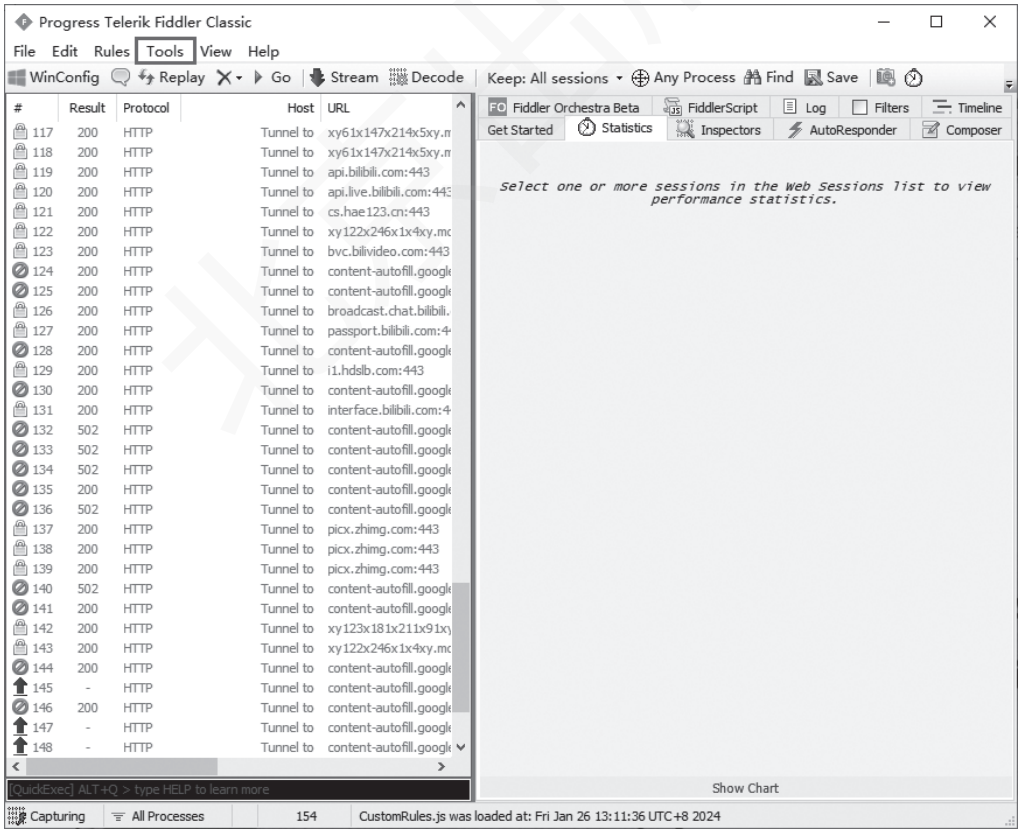


图 1-8　配置 Fiddler 工具

单击图 1-8 中“Tools”菜单下的“Options”按钮进入工具配置界面，如图 1-9 所示。

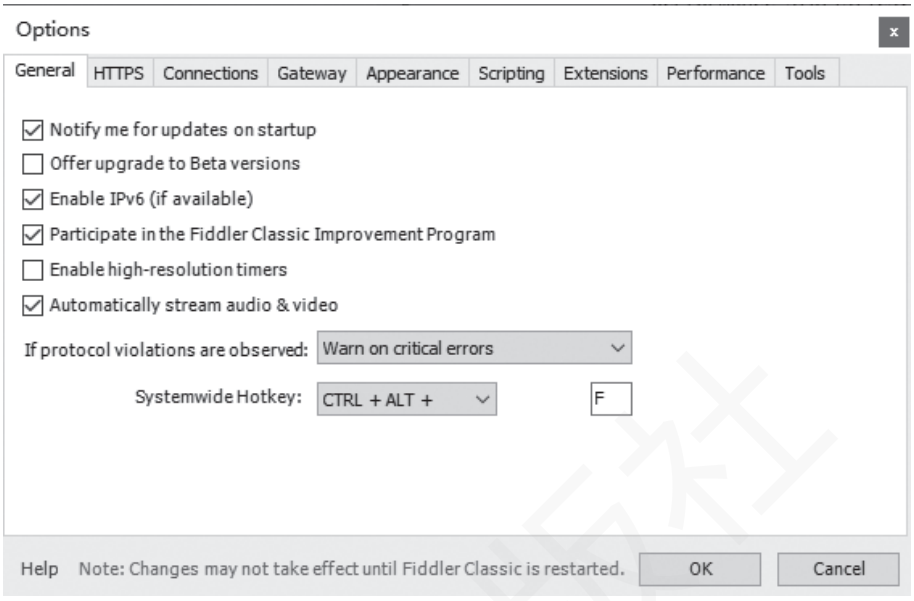


图 1-9 工具配置界面

选择“Connections”选项卡后，进行端口号的配置，如图 1-10 所示。

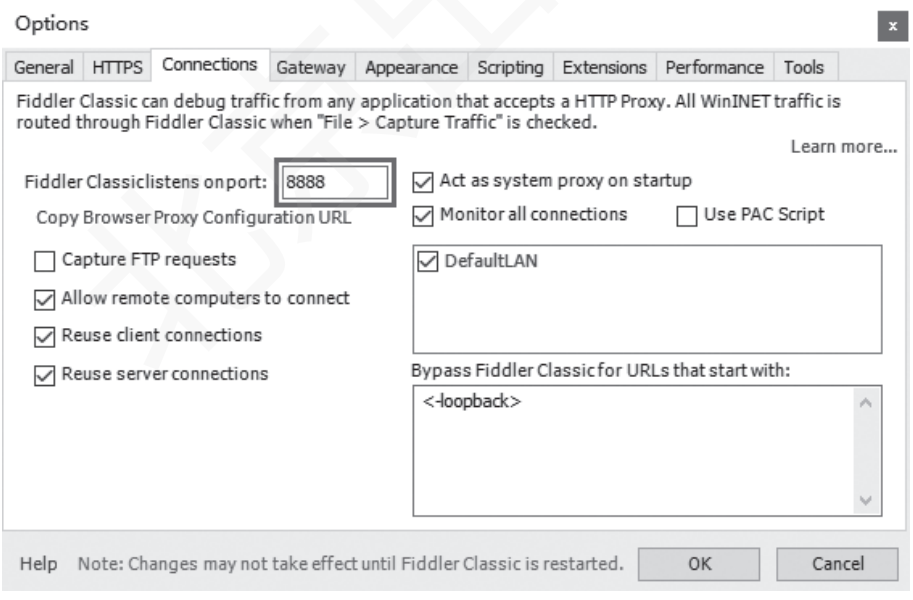


图 1-10 配置端口号

（四）配置手机

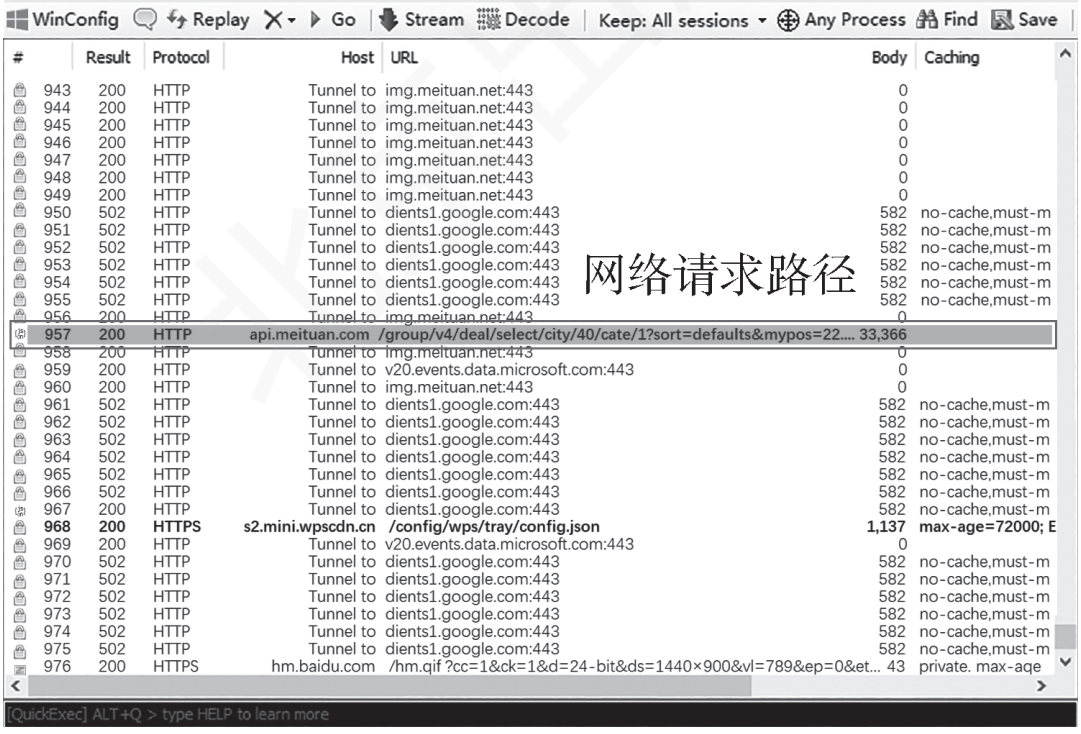
由于爬取的是手机 App 数据，因此需要在同一局域网内进行手机网络的配置。进入手机 Wi-Fi 修改界面，设置手动代理并进行主机 IP 和端口号的配置，如图 1-11 所示。

图 1-11　配置手机

（五）分析 App 页面

配置完成后，即可使用当前手机打开需要爬取的 App，这里使用的是美团 App。

找到需要爬取的页面后，在 Fiddler 抓包工具页面中会获取到当前 App 请求网络的路径，单击路径后即可查看当前 App 的相关信息，如图 1-12 和图 1-13 所示。



#	Result	Protocol	Host	URL	Body	Caching
943	200	HTTP	Tunnel to	img.meituan.net:443	0	
944	200	HTTP	Tunnel to	img.meituan.net:443	0	
945	200	HTTP	Tunnel to	img.meituan.net:443	0	
946	200	HTTP	Tunnel to	img.meituan.net:443	0	
947	200	HTTP	Tunnel to	img.meituan.net:443	0	
948	200	HTTP	Tunnel to	img.meituan.net:443	0	
949	200	HTTP	Tunnel to	img.meituan.net:443	0	
950	502	HTTP	Tunnel to	dients1.google.com:443	582	no-cache,must-m
951	502	HTTP	Tunnel to	dients1.google.com:443	582	no-cache,must-m
952	502	HTTP	Tunnel to	dients1.google.com:443	582	no-cache,must-m
953	502	HTTP	Tunnel to	dients1.google.com:443	582	no-cache,must-m
954	502	HTTP	Tunnel to	dients1.google.com:443	582	no-cache,must-m
955	502	HTTP	Tunnel to	dients1.google.com:443	582	no-cache,must-m
956	200	HTTP	Tunnel to	img.meituan.net:443	0	
957	200	HTTP	api.meituan.com	/group/v4/deal/select/city/40/cate/1?sort=defaults&mypos=22....	33,366	
958	200	HTTP	Tunnel to	img.meituan.net:443	0	
959	200	HTTP	Tunnel to	v20.events.data.microsoft.com:443	0	
960	200	HTTP	Tunnel to	img.meituan.net:443	0	
961	502	HTTP	Tunnel to	dients1.google.com:443	582	no-cache,must-m
962	502	HTTP	Tunnel to	dients1.google.com:443	582	no-cache,must-m
963	502	HTTP	Tunnel to	dients1.google.com:443	582	no-cache,must-m
964	502	HTTP	Tunnel to	dients1.google.com:443	582	no-cache,must-m
965	502	HTTP	Tunnel to	dients1.google.com:443	582	no-cache,must-m
966	502	HTTP	Tunnel to	dients1.google.com:443	582	no-cache,must-m
967	200	HTTP	Tunnel to	dients1.google.com:443	582	no-cache,must-m
968	200	HTTPS	s2.mini.wpscdn.cn	/config/wps/tray/config.json	1,137	max-age=72000; E
969	200	HTTP	Tunnel to	v20.events.data.microsoft.com:443	0	
970	502	HTTP	Tunnel to	dients1.google.com:443	582	no-cache,must-m
971	502	HTTP	Tunnel to	dients1.google.com:443	582	no-cache,must-m
972	502	HTTP	Tunnel to	dients1.google.com:443	582	no-cache,must-m
973	502	HTTP	Tunnel to	dients1.google.com:443	582	no-cache,must-m
974	502	HTTP	Tunnel to	dients1.google.com:443	582	no-cache,must-m
975	502	HTTP	Tunnel to	dients1.google.com:443	582	no-cache,must-m
976	200	HTTPS	hm.baidu.com	/hm.gif ?cc=1&ck=1&d=24-bit&ds=1440×900&vl=789&ep=0&et...	43	private. max-age

图 1-12　App 请求网络路径

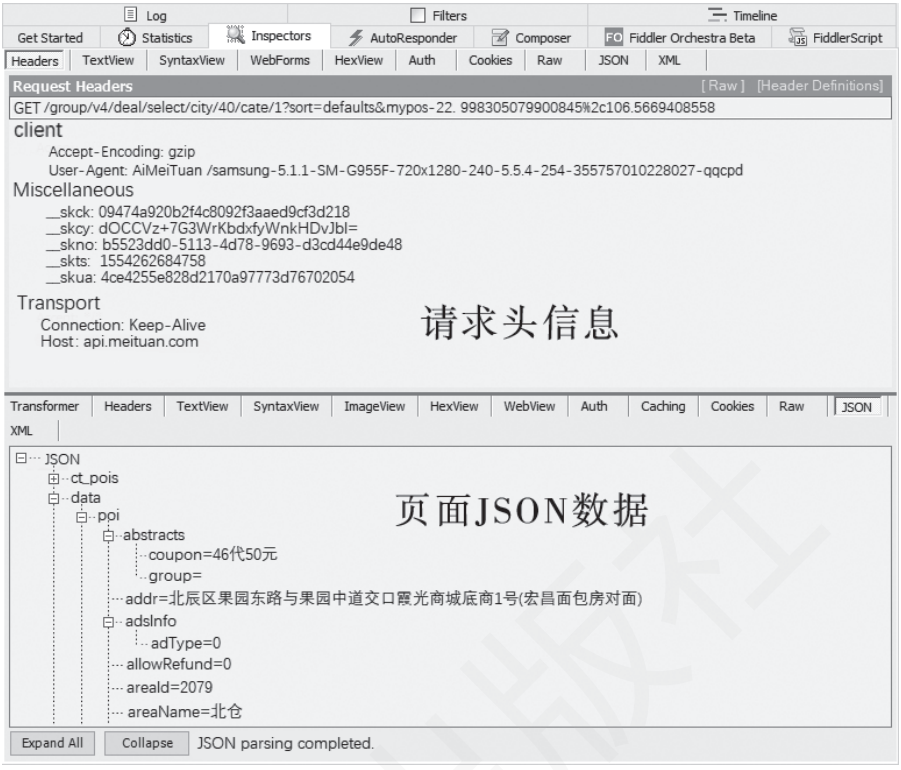


图 1-13 App 相关信息

（六）编辑代码

基本配置和信息获取完成后即可进行代码的编辑。将上面获取的相关请求头信息填入代码相应的位置，将爬取路径放入请求方法中进行页面内容的请求，通过 JSON 信息的分析爬取需要的页面信息，如有需要可将信息保存到本地文件中，代码如下。

```
# 引入 requests 库
import requests
def main():
    # 定义请求头
    headers={
        # 将 Fiddler 右上方的内容填在 headers 中
        "Accept-Charset":"UTF-8",
        "Accept-Encoding":"gzip",
        "User-Agent":"AiMeiTuan/OPPO-5.1.1-OPPO R11-1280x720-240-5.5.4-
254-8661740 10228027-qqcpd",
        "Connection":"Keep-Alive",
        "Host":"api.meituan.com"
    }
    # 循环请求数据
    for i in range(0,100,15):
        # 右上方有个 get 请求，将 get 后的网址赋给 heros_url
```

```
            heros_url="http://api.meituan.com/group/v4/deal/select/city/40/
cate/1?sort=defaults&mypos=33.99958870366006%2C109.56854195330912&has
Group=true&mpt_cate1=1&offset="+str(i)+"&limit=15&client=android&utm_
source=qqcpd&utm_medium=android&utm_term=254&version_name=5.5.4&utm_
content=866174010228027&utm_campaign=AgroupBgroupCOEOGhomepage_
category1_1__al&ci=40&uuid=704885BFB717F2C01E511F22C00C57BCF67FBCCB6E51D4E
E4D012C5BE0DCAFC2&msid=866174010228027155109995284 8&__skck=09474a920b2f4c
8092f3aaed9cf3d218&__skts=1551100036862&__skua=4cc9b4c45a5fd84d9e60e187fa
bb4428&__skno=6b0f65d3-0573-483c-a0c0-68a16fdldda7&__skcy=ylVLNnkSr%2BWmTK
Ufgw%2BL6Ms21sg%3D"
            # 美食的列表显示在 json 格式下
            res=requests.get(url=heros_url,headers=headers).json()
            # 打印列表
            for i in res["data"]:
                print(i["poi"]["name"])
                print(i["poi"]["areaName"])
                print(i["poi"]["avgPrice"])
                print(i["poi"]["avgScore"])
                print("++++++++++++++++++++++++++++++++++++=")
    if__name__=="__main__":
        main();
```

运行代码，效果如图 1－14 所示。

```
梨花炭团烧肉丼（天河城店）
和平路
39
5
++++++++++++++++++++++++++++++++++
彤德菜火锅（天河城店）
和平路
62
5
++++++++++++++++++++++++++++++++++
屯老二农家铁锅炖（张家窝店）
天津南站
73
5
++++++++++++++++++++++++++++++++++
屯老二农家铁锅炖（双街店）
双街
69
5
++++++++++++++++++++++++++++++++++
火炉火烤肉·芝士排骨（世纪都会店）
滨江道
81
5
++++++++++++++++++++++++++++++++++
新辣道（荔隆店）
北郑/泉州路北
81
```

图 1-14　爬取数据效果图

项目评价

以小组为单位，配合指导老师完成表 1-2 所示的项目评价表。

表 1-2 项目评价表

项目名称	评价内容	分值	评价分数		
			自评	互评	师评
职业素养考核项目（20%）	考勤、仪容仪表	5			
	责任意识、纪律意识	5			
	团队合作与交流	10			
专业知识考核项目（40%）	互联网数据的来源、特征	10			
	数据采集的概念、步骤及方式方法	10			
	网络爬虫的概念、结构、组成、类型	10			
	网络爬虫的相关技术	10			
专业能力考核项目（40%）	举例说明数据采集的典型应用	20			
	能够使用 requests 模块进行手机端数据的爬取	20			
合计：综合分数 _____ 自评（20%）+ 互评（20%）+ 师评（60%）		100			
综合评语		教师（签名）：			

思考与练习

一、选择题

1. 互联网数据来源不包括（　　）。

A. 百科知识库　　B. 新闻网站

C. 评论信息　　D. 国家统计局

2. 下列不属于线上行为数据的是（　　）。

A. 页面数据　　B. 交互数据

C. 表单数据　　D. 机器数据

3. 互联网大数据采集的特征是（　　）。

A. 来源单一　　B. 结构单一

C. 数据类型丰富　　D. 以上都是

4. 主要的数据预处理技术不包括（　　）。

A. 数据中心传输　　B. 数据整合

C. 数据清洗　　D. 冗余消除

5. 关于网络爬虫的控制节点和爬虫节点，下列说法不正确的是（　　）。

A. 网络爬虫可以有多个控制节点，每个控制节点下又可以有多个爬虫节点

B. 控制节点之间可以互相通信

C. 控制节点和其下的各爬虫节点可以互相通信

D. 不同控制节点下的各爬虫节点亦可以互相通信

6. 利用贝叶斯分类器，根据整个网页文本和链接文本对超链接进行分类，对每个链接计算出重要性，从而决定链接的访问顺序。该策略属于（　　）。

A. 基于内容评价的爬行策略

B. 基于链接结构评价的爬行策略

C. 基于增强学习的爬行策略

D. 基于语境图的爬行策略

二、填空题

1. 大数据特征包括数据体量巨大、_____、_____、数据产生和处理速度快。

2. 数据采集过程可分为数据收集、_____、_____ 三个步骤。

3. 按照数据的形态，可以把数据分为 _____ 和 _____ 两种。

4. 网络爬虫由 _____、_____、_____ 构成。

5. 网络爬虫可以分为通用网络爬虫、_____、_____、_____ 类型。

三、简答题

1. 什么是数据采集？

2. 简述传统数据采集与互联网大数据的数据采集的区别。

3. 数据采集的方法有哪些？

4. 简述深度优先策略和广度优先策略的异同。

四、操作题

通过网络查找一些数据采集违法的案例并进行分析讨论，讲一讲这些案例给你带来的启发。

1. 全班分成若干个小组，每组 4～6 人。

2. 以小组为单位完成案例收集与整理。

3. 每个小组提交一份案例报告（Word 或 PPT 形式均可）并汇报。