基于大数据的 51job 网站大学生大数据岗位数据分析

赵红波 张桂花

四川大学锦城学院计算机与软件学院 四川 成都 611731

【摘 要】信息时代的到来，促进了数据的快速增长，当下时代数据的开放性使得获取数据变得尤为方便；而数据的海量性，使得大学生对于数据的辨析变得尤为困难。便利性、时效性、覆盖性强等特点已经逐渐成为当下大学生最受欢迎的寻找职位方式，而数据的海量性使得大学生通常不能完整的分析数据。本文通过对 51job 网站进行数据爬取，采用 python 程序对数据进行清洗，然后使用 Hadoop 核心组件 HDFS 进行存储，使用 Spark 技术对其进行分析和处理，其结果以可视化方式进行展示，对大学生提供一份可靠的数据。

【关键词】大数据；HDFS；大学生；Spark

1 绪论

1.1 背景

网络的发展速度让我们感到震惊，当今人类社会已经进入了信息时代，在各种信息的爆炸式呈现中，大数据悄然进入了我们的视线。如今大数据早已与各行各业的发展息息相关。网站招聘由于其便利性、时效性、覆盖性强等特点已经逐渐成为当下大学生最受欢迎的寻找职位方式。大学生在对网站的简单了解后，在网页的搜索栏输入关键字即可寻找到所需的大量相关数据。

1.2 研究意义

获取的数据信息量大、种类繁多、实时更新、价值密度低都符合当下大数据的特征。虽然数据可轻易获得，但后续仅凭人力我们很难从这些海量数据中提炼出符合我们要求的有价值的信息，例如各个地区的各类岗位的平均薪资、对于同一类岗位出现次数最高的需求、各个城市对于岗位的需求分布情况，人力往往无法完成这样的海量分析，而大数据可以精确的分析出所有的数据，最终呈现出一份详细的、完整的、准确的报告。大数据的分析统计是实时的、全面的，充分体现了数据的价值，才能对大学生进行准确的指导，了解了自己的不足后，才能有针对性的在校期间进行相应的理论学习和技能培训，在未来的就业竞争中更具有核心竞争力，提高大学生应聘成功率。

2 项目流程介绍

本文的数据采集在 51job 招聘网中进行，首先在登录了 51job 网站后，在搜索栏输入我们想要了解的职位后（本文搜索大数据相关的职业），分析网页数据结构特征，编写 Python 网络爬虫程序。爬取初始 url，并根据初始 url 爬取新的 url 和每一页的数据，重复操作，完成爬取数据。将爬取的数据在 Pycharm 中通过逻辑代码删除空值（NAN）、无关的职位信息、错位的信息、重复和错误的数据，即可完成数据的清洗工作，并且在数据上传到 HDFS 前进行数据预处理。Python 网络爬虫程序爬取到的数据保存在本地的“message.xls”表格中，将表格通过 Xshell 上传到 HDFS。因为 Spark 也是在 HDFS 上面进行操作，所以直接读取 HDFS 上的文本，通过 Spark 中的 RDD 算子进行数据的分析和统计。整个项目的流程如图所示。

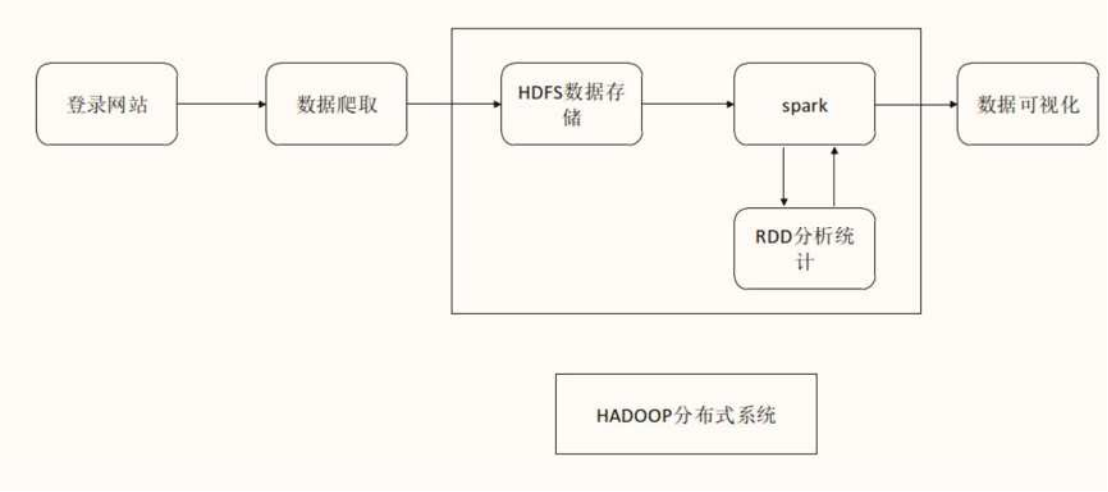

图 1 数据流程图

Fig. 1 Data flow chart

3 技术实现

3.1 数据采集

大数据的基础就是数据的海量性，首先需要获得足够多的数据，本文通过编写 Python 网络爬虫程序，在 51job 招聘网上爬取大数据职位相关信息，为后续的数据处理做准备。下图为爬虫程序流程图。

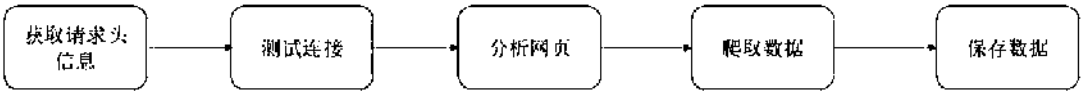

图 2 网络爬虫流程图

Fig. 2 Web crawler flow chart

3.1.1 数据网页分析

首先登陆 51job 网站，在主页面的搜索框中搜索大数据，使用谷歌浏览器 Chrome 的开发者工具，分析网页的数据结构，了解请求的地址、类型、请求头、请求参数等信息,为下一步编写网络爬虫程序做好准备[1]。在分析了网页数据结构后，对标签中的定位采用 XPath 插件，XPath 是一门在 XML 文档中查找信息的语言，同样也适用于 HTML 文档的搜索。

3.1.2 导入库并对目标网址连接测试

首先在Pycharm中导入爬虫程序所需要的库,例如:requests、re、xlwt。在 Chrome 的开发者工具中得到了请求的地址、类型、请求头、请求参数等信息后，模拟浏览器。通过req库中的get(url)函数的返回值(200表示成功）测试是否能连接目标网址，爬取网站信息。

3.1.3 构造 URL

爬取成功后在 Chrome 的开发者工具中分析网页数据特征，然后在网址栏中分析每一页网址共同点，封装函数 getUrl，分别爬取每一页中新的网址和有效数据。首先获取初始的 URL，根据初始的 URL 爬取页面并获得新的 URL，将新的 URL 放入 URL 队列中，从 URL 队列中读取新的 URL，并根据新的 URL 爬取网页，同时从新的页面获取 URL，重复操作，满足爬虫设置的条件时，停止爬取。getUrl 函数中采用网址拼接，将相同部分设置为常量，不同部分用 xpath 在网页中寻找相关网址部分段，循环生成新的网址，直到结束。下图为 URL 构造表格。

表 1 URL 构造表格

Table 1 URL construction table

函数传参	def getUrl(page,item):
参数转码	result = urllib.parse.quote(item)
url1（固定	ur2 = ‘https://search.51job.com/list/000000,000000,0000,00,9,99,’
url2(拼接)	ur2 = ‘https://search.51job.com/list/000000,000000,0000,00,9,99,’

3.1.4 正则表达式提炼网页数据

封装函数 getInformation，在提取数据特征后 getInformation 函数中编写正则表达式提取数据，爬取数据字段依次为：‘序号’，‘职位’，‘公司名称’，‘公司地点’，‘公司性质’，‘薪资’，‘学历要求’，‘工作经验’，‘公司规模’，‘公司福利’，‘发布日期’采用 for 循环全部爬取并保存在本地新建的“message.xls”表格中。

3.2 数据清洗

数据清洗和后续的数据预处理是为了 Spark 对存储在 HDFS 上的文档分析和统计更为方便。直接从网页中采集的数据类型繁杂多样，数据中夹杂 着不完整、重复以及错误的数据，而且大多数数据是我们分 析工作所不需要的数据，如果直接进行分析的话，会影响分析效率，甚至会导致分析结果的错误，最终会导致决策的错误[2]。所以在后续的数据存储、分析统计等操作前对数据的清洗尤为关键。

在本地打开“message.xls”表格，查看整个表格，观察数据结构和错误类型。在文本中发现有空值、缺失值、重复值、错位值、错误值四种需要处理的情况。在 Pycharm 中读入“message.xls”表格，当检测到表格中行和列中数据的有空值，直接删除整个行和列。代码如下：

```
data = pd.read_excel('51job',sheet_name='Job')
a = pd.DataFrame(data)
b = a.dropna(anxis=0,how='any')
```

通过 data.drop_duplicates() 函数发现有两行即多行中的数据重复，直接删除。我们在爬取了数据后发现爬取的职位栏中会出现一些其他职位的数据，所以我们通过循环直接将职位栏中不是大数据的职位数据一行数据全部删除。在薪资列中出现单位不一样，通过正则表达式提取薪资列的 Int 类型数据将其换算为统一单位。在后期的数据可视化展示，我们需要准备数据的分类和统计，通过 For 循环在公司地点列将所有大城市地点爬取存放在列表中，然后去重保存在新的列表中。根据列表的长度创建对应的文件，以列表中保存的数据为

文件名。在整个表格中把各个地点的数据通过 For 循环将数据以地点分类到创建的各个文件中。

3.3 数据存储

3.3.1HDFS 存储数据

采集的数据由于相对较大，我们采用 Hadoop 分布式系统基础架构下的 Hadoop 分布式文件系统（HDFS）进行存储，HDFS 支持海量数据的存储，允许用户把成百上千的计算机组成存储集群，其中的每一台计算机为一个节点，采用 Master/Slav 结构模型进行管理 [3]。Hadoop 分布式文件系统提供了高容错性和恢复机制，具有其高可靠性的特点；适合处理大规模数据；采用了流式文件访问模式，支持一次写入，多次读出，可保证其数据的一致性。

3.3.2 上传数据

在本地保存的数据通过 Xshell 软件上传到 HDFS。在下载好了 Xshell 软件后打开工具，连接到服务器，输入服务器的 IP、账号、密码，使用 yum -y install lrzsz 命令安装 lrzsz 工具，其后通过 rz 命令完成文件上传。

3.4 数据统计

3.4.1 采用 Spark 原因

Spark 由 Scala 语言实现，支持 Python、Java、Scala 等语言开发，可以和其他大数据工具如 Hadoop、Kafka 等很好地整合。Spark 是基于内存计算的，且具有易操作的特点，能够快速、简洁、高效的进行并行化应用的开发 [4]。Spark 是基于内存计算的大数据分布式框架，所以有着低延迟的复杂分析。优于 MapRuduce，由于 Spark 中间输出结果可以保存在内存中，所以可以减少读写 HDFS 的次数。

3.4.2 采用 Spark 分析、统计数据

首先使用 Spark 在 HDFS 上读取文件，val rdd = sc.textfile(“文件路径 / 文件名”)。读取后的文件使用 RDD 算子中的 groupByKey（）对文件每条数据进行统计，lines.flatMap(line => line.split(“,”)).map(word =>(word,1)).reduceBuKey((a,b)=> a+b)。RDD 首先把文件一行一行的读进来，通过 map（）一行数据生成一个 Array（），Array（）中的数据用逗号分开。再把 map（）操作得到 Array（）中的每个元素形成键值对。最后通过 reduceByKey（）聚合统计数据。操作流程如下图所示。

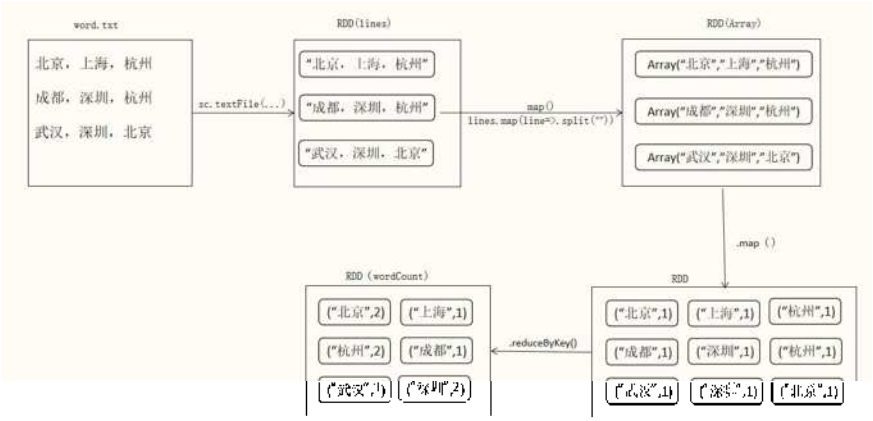

图 3RDD 操作流程图

Fig.3 SPARKRDD operation flowchart

3.5 数据可视化

在面对庞大的、复杂的数据，人们往往无法快速捕捉到数据的价值。所以往往在得到了我们需要借助可视化工具完成对数据的清晰表达，有效传达信息。可视化的基本思想就是用更易理解式来表达数据、描述问题，即通过对数据的进一步加工和转换来达到更直观地进行信息启示的目的，其最大的优点是可以降低认知成本，提高认知效率 [5]。在通过 Spark 中 RDD 算子统计分析了数据，在 Pycharm 进行可视化。下图为一部分可视化展示。

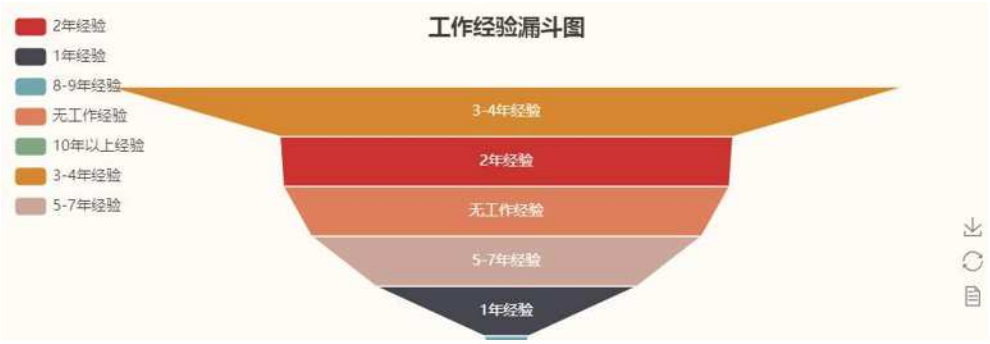

图 4 工作经验漏斗图

Fig.4 Work experience funnel plot

结论

IT 行业是一个飞速发展的行业，大数据在其中的地位不言而喻。信息爆炸式的增加，给人来带的不仅是便利的生活，也带了更大的挑战：如何快速的辨析数据。本文从大学生网络招聘实际问题入手，通过爬虫程序采集数据、Python 程序清洗和预处理数据、HDFS 储存数据、Spark 分析数据、最后在 Pycharm 中可视化数据。通过可视化的展示中学历要求：本科占据 70% 以上，初中及以下最少；工作经验 3-4 年人数要求最多，10 年以上经验要求最少；平均薪资排名前五的城市分别为：北京、深圳、上海、杭州、广州。通过以上展示数据可以对大学生提供价值较高的参考。

【参考文献】

[1] 朱永忠 . 基于大数据技术的大学生就业分析系统的研究 [J]. 现代信息科技 ,2020,4(18):128-130+136.

[2] 朱永忠 . 基于大数据技术的大学生就业分析系统的研究 [J]. 现代信息科技 ,2020,4(18):128-130+136.

[3] 邱春红 . 基于 Hadoop 的农产品追溯系统框架研究 [J]. 电子测试 ,2021(09):74-76.

[4] 张力元 . 基于 Spark 的混合模式电影推荐系统研究与实现 [D]. 重庆大学 ,2018.

[5] 魏红君 . 市场监管业务数据可视化平台研究与实践 [J]. 科技风 ,2021(13):101-102.