

**课程报告**

**课程名称：大数据导论**

**专业班级：**

**学 号：**

**姓 名：**

**指导教师：**

**报告日期： 2020年1月2日**

**计算机科学与技术学院**

**目 录**

[基于Hadoop的知乎社群分析 3](#_Toc28888881)

[1.1 整体方案简介和分级任务简介 3](#_Toc28888882)

[1.2 数据集说明 3](#_Toc28888883)

[1.3 工具说明 3](#_Toc28888884)

[1.4 问题挑战 3](#_Toc28888885)

[1.5 成果展示 3](#_Toc28888886)

[1.6 心得体会与总结 3](#_Toc28888887)

基于Hadoop的知乎社群分析

* 1. 整体方案简介和分级任务简介

整体方案简介：

爬取知乎用户数据，清洗后交由系统级和测试级进行测试分析。

分级任务简介：

本小组负责数据的爬取和清洗工作。

初始目标选取为NBA网站球员的个人数据：<http://www.stat-nba.com/query.php?page=1&QueryType=game&order=1&crtcol=pts>。后续考虑到数据量级(网站爬取下来的的数据规模过小)及分析的实际需求，改为知乎网站，从知乎API接口[https://www.zhihu.com/api/v4/members/{user}?include={query}](https://www.zhihu.com/api/v4/members/%7buser%7d?include=%7bquery%7d)提取json数据后存储到Mongodb数据库中，用Studio-3T导出为csv格式。

数据清洗：将爬取的数据进行一定的清洗工作，从而方便后续系统级和分析级展开更深入的分析工作。

任务分工：

朱志成：数据爬取

李晓宇：数据清洗

* 1. 工具说明

数据爬取：

数据来源：知乎用户信息

使用工具：Python-Scrapy爬虫框架，Mongdb数据库，Studio-3T可视化

工作环境：Windows10；Python 3.6；Mongodb；16G内存

数据清洗：

数据来源：爬取到的数据

使用工具：Kettle

工作环境：Windows10；32G内存；处理器i7-6600U

* 1. 具体流程

数据爬取：

数据来源：知乎API：[https://www.zhihu.com/api/v4/members/excited-vczh?include={user\_query}](https://www.zhihu.com/api/v4/members/excited-vczh?include=%7buser_query%7d)

通过次接口可以获得的数据字段如图所示，共有五十项左右，图1所示为知乎请求个人信息时的各个字段。

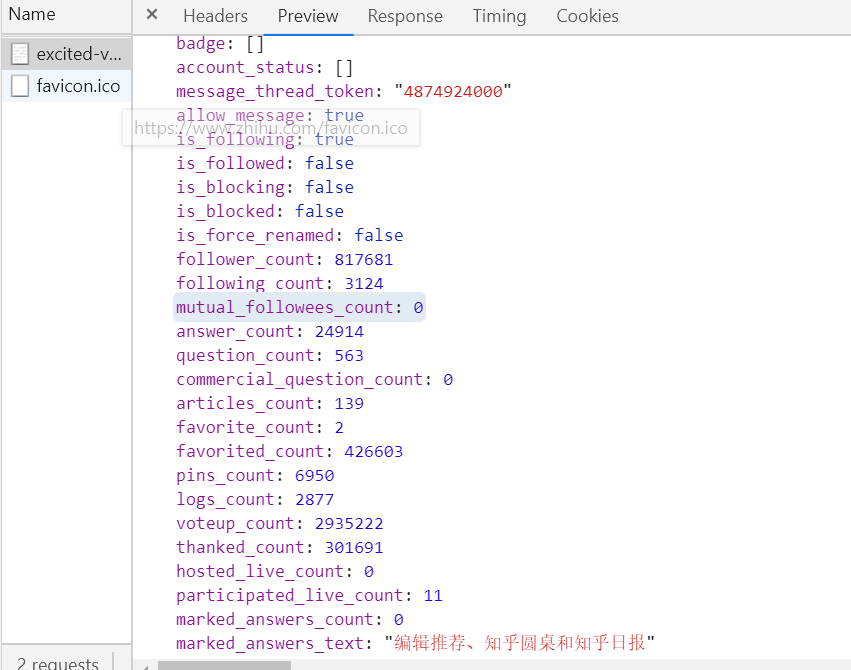


图1 API json数据

从中选取了37个意义较为明确的字段进行了解析。如图2 ，包括：

Id（用户标识），name（用户昵称），url\_token（URL别名），url（完整url），headline （用户介绍），badge（徽章），answer\_count（回答数），question\_count（提问数），commercial\_question\_count（商业提问数）， articles\_count（文章数），thanked\_count（感谢数），voteup\_coun（赞同数），favorited\_count（收藏数），favorite\_count（专业认可数），follower\_count（粉丝数），following\_count（关注数），following\_question\_coun（关注问题数），following\_topic\_count（关注话题数），following\_favlists\_count（关注收藏夹数），following\_columns\_count（关注专栏数 ），participated\_live\_count（赞助live数），hosted\_live\_count（主持live数），avatar\_url （头像url），description（个人简介），gender（性别 | 1为男性，0为女性，-1为未知），cover\_url （背景url），pins\_count（想法数），type （用户类型 | 一般为people），educations（教育经历），employment （职业经历），locations（居住地）



图2 Items.py

以一个用户作为爬取起点，后续跟踪其粉丝列表，以广度优先遍历的方式逐个查询，为了保证数据的全面性，共爬取了10次，每次爬取时间均控制在25分钟左右，总时间为5h左右。

配置好Mongodb数据库后，将item中的数据存储到mongodb数据库中时，在Scrapy中进行了一个去重检测，保证同一个csv下不会有相同的用户项出现

数据清洗：

1. 原始数据特征：

（1）有很多无用或出现乱码的字段，例如URL字段

（2）多个表项之间会有重复的用户数据

（3）部分字段有很多无用和冗余的信息，如教育经历和学校字段混杂着相关的ID和URL

2. 清洗流程：

（1）先将原始的数据导入到软件中，定义相关字段的数据类型，并去掉一些无用字段

（2）编写Java脚本，从包括无用冗余信息的字段中提取出关键字，输出到新的字段中

（3）基于用户ID进行数据的排序，启动合并排序功能将每个单独的数据集合并成完整的数据集

（4）启用去重操作，在ID有序的完整数据集中去掉重复ID的用户

（5）将完整数据集分割导出到一定数量的输出文件中，方便后续数据的分布式处理

具体的Kettle流程如下所示：

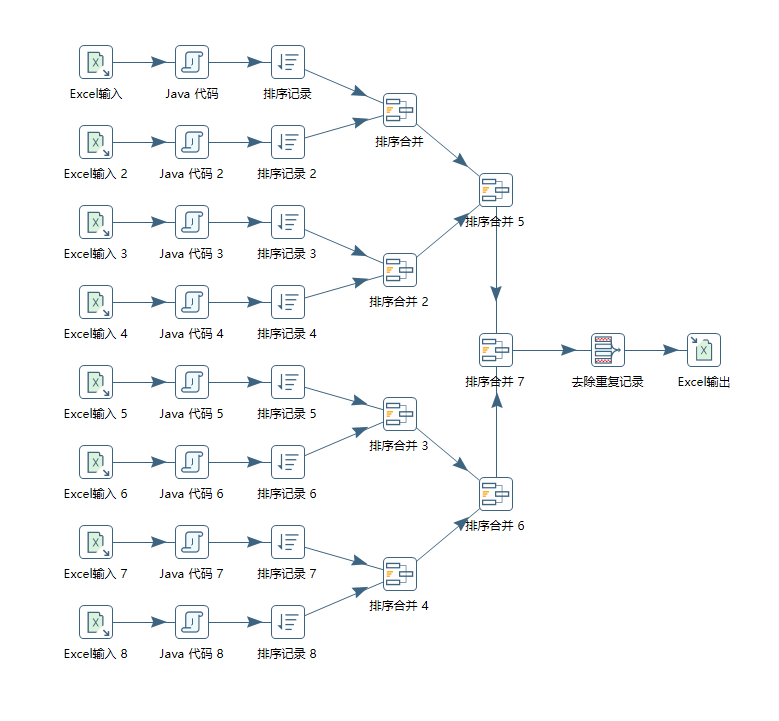


图3 Kettle流程图



图4 Java脚本图

* 1. 问题挑战

数据爬取：

问题2：数据的大小无法保证

解决方案：最初爬取的篮球网站数据，结果数据的规模过小，只有数MB，考虑 到一般网站的数据量确实无法达到要求，后改为爬取知乎用户的信息数据

问题2：如何保证所爬取用户的全面性？不能局限于某一个领域

解决方案：选取了多个不同领域的大V作为数据爬取的起点，将其粉丝群体作为后续起点，如图5所示。

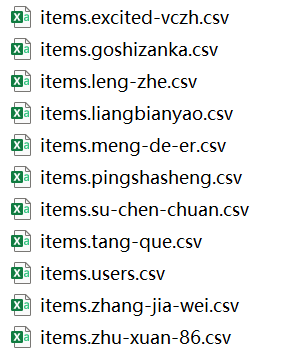


图5 爬取结果

数据清洗：

问题1：数据条目众多，Kettle进行合并去重时速度缓慢。

解决方案：利用归并排序的思路，将文件二分合并并加以去重，节省了许多没有必要的排序和合并操作，加快了Kettle处理数据的速度。

问题2：数据条目庞大，难以整合到一个Excel表中。

解决方案：将输出文件分割为数个小的输出流，重新输出到若干个Excel文件里。

* 1. 成果展示

数据爬取：图10为爬取后存储到Mongodb中的数据。

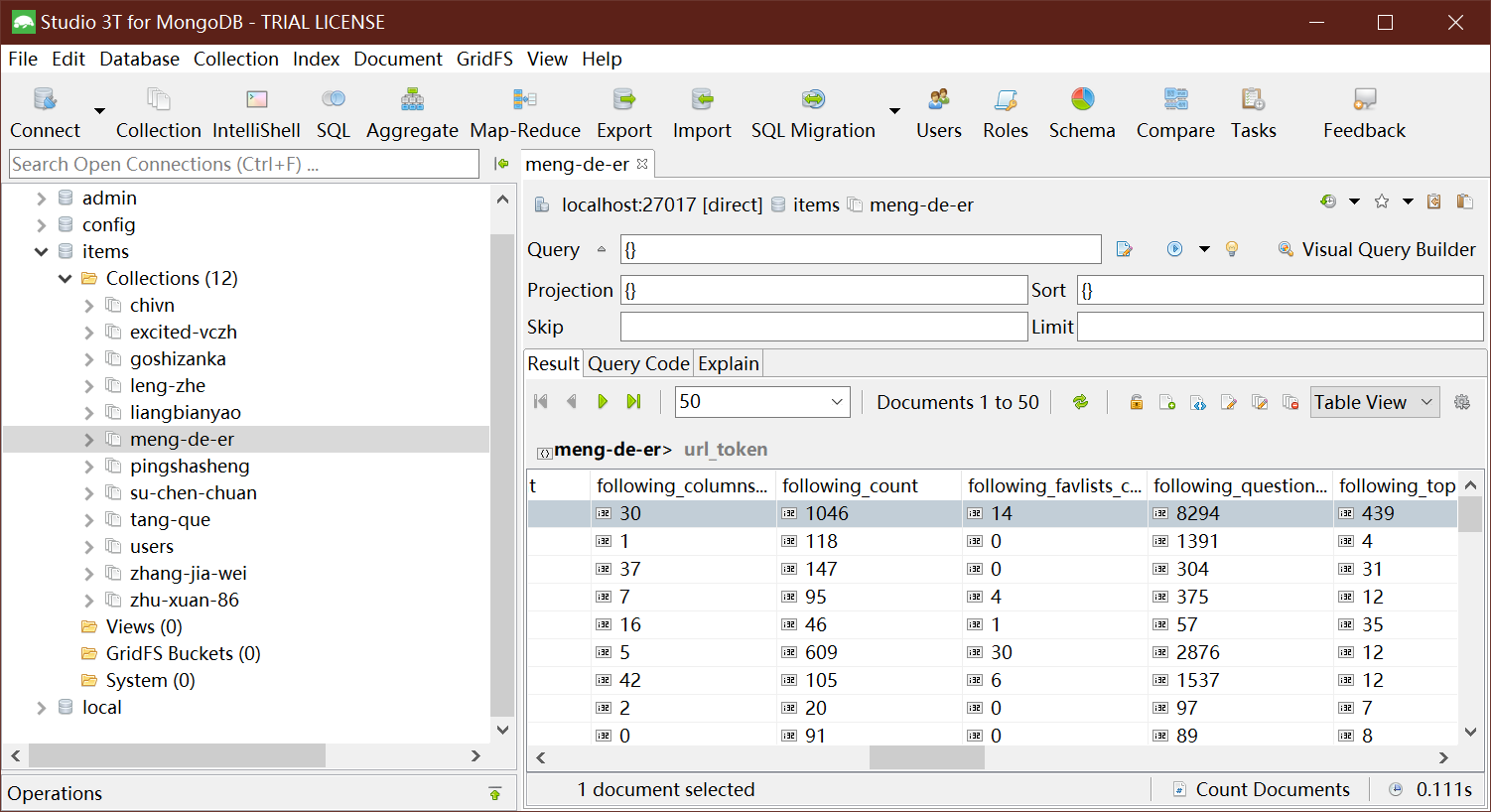


图6 爬取数据展示

导出为csv文件后数据总大小为140M，用户量接近20w，后交由数据清洗。



图7 爬取数据规模

数据清洗：从下图可以看出education和badge等字段都已经提取出了关键信息，且许多重复的条目都被清除。

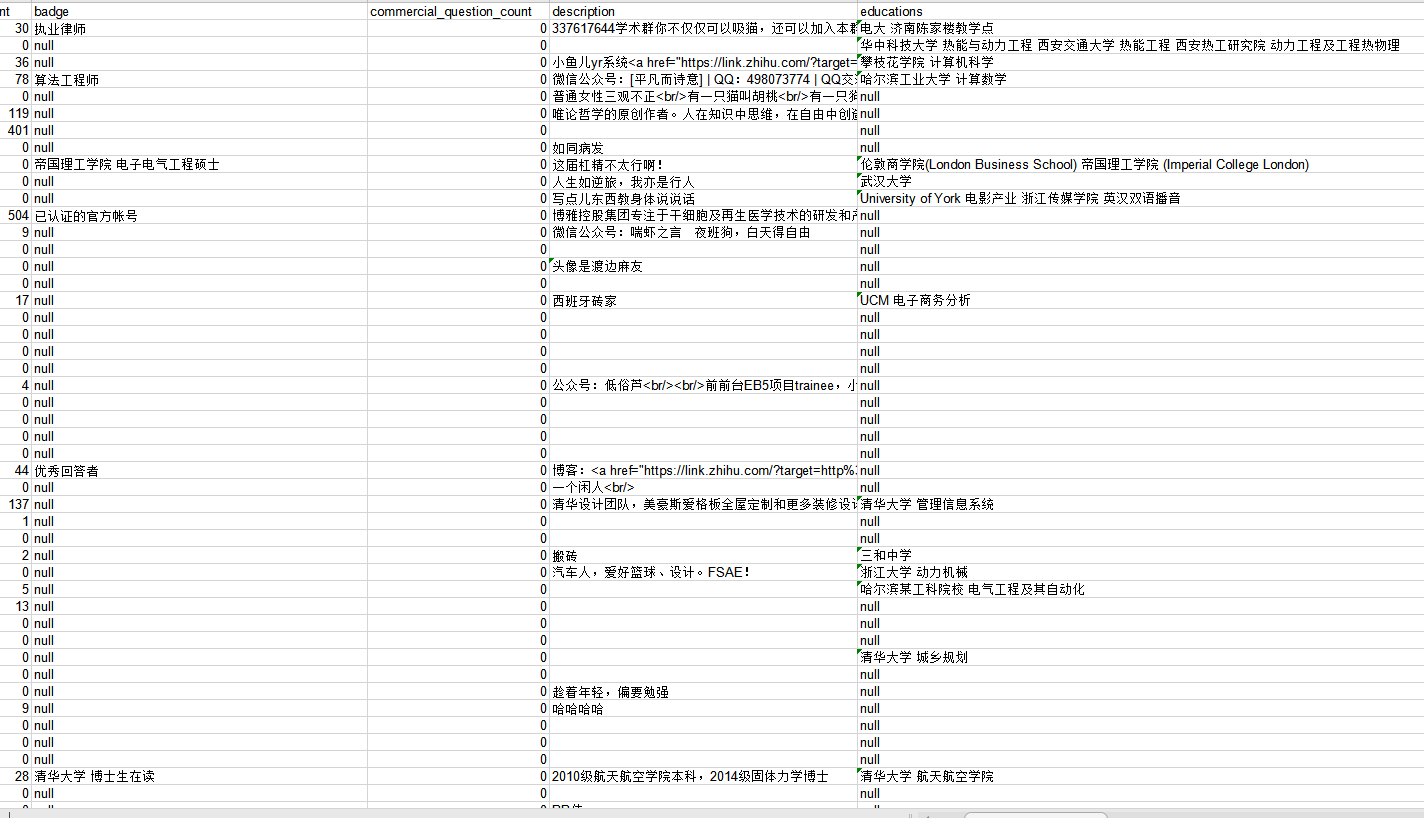


图8 清洗后的数据展示

* 1. 心得体会与总结

数据爬取：本次数据爬取经历了两个阶段，第一阶段用自己编写的爬虫爬取篮球网站的数据，第二阶段爬取知乎数据使用了Scrapy框架，相比之下速度有了一定程度上的提升。通过本次数据爬取工作，我对于爬虫的工作原理以及常用爬虫框架的使用有了一定的了解，能够满足基本的数据爬取要求，并且有了自己编写爬虫的能力。但是本次爬取中做了较多的无用功，主要是由于数据源选取不当，事前为做好数据量级的估计工作，导致实际结果与预期出现了较大偏差。

数据清洗：通过这次的数据清洗操作，我大致接触了Kettle的相关操作，熟悉了Kettle处理数据的流程，并且能够自己编写新的Java脚本来方便Kettle的数据处理。这些内容开拓了我的眼界，使我接触到了ETL，且对市面上其它ETL工具也有了大致的认识。