MapReduce 大数据实验-金庸的江湖

吴刚，吴德亚，吴晗，温宗儒

2019 年 7 月 22 日

目录

1 实验要求及目标

通过对“金庸的江湖——金庸武侠小说中的人物关系的挖掘”，来学习与掌握 MapReduce 程序设计。通过本课程设计的学习，可以体会如何使用 MapReduce 完成一个综合性的数据挖掘任务，包括全流程的数据预处理、数据分析、数据后处理等。

本次实验的具体流程如下:

(1) 数据预处理：利用分词技术将文章中人物名字保留下来，其余去掉。

(2) 特征抽取：人物同现统计。

(3) 特征处理：人物关系图构建与特征归一化。

(4) 数据分析：基于人物关系图进行 PageRank 计算或者标签传播。

(5) 数据可视化及分析。

2 任务分配

161220137 吴刚 ：任务三，任务五，数据可视化及相应报告书写

161220135 吴德亚 ：任务四，相应报告书写及 latex 报告转换

161220138 吴晗 ：任务一及相应报告书写

161220133 温宗儒 ：任务二及相应报告书写

3 实验环境

(1)Ubuntu1604

(2)Intellij IDEA 2018

(3)JDK 1.7_79

(4)hadoop 2.7.1

4 实验过程

4.1 一 数据预处理

本任务的主要工作是从原始的金庸小说文本中，抽取出不人物互动相关的数据，而屏蔽掉不人物关系无关的文本内容，为后面的基于人物共现的分析做准备。

4.1.1 任务简述

数据输入：1. 全本的金庸武侠小说文集（未分词）；2. 金庸武侠小说人名列表。

数据输出：分词后，仅保留人名的金庸武侠小说全集。

此处按照要求，对每一个小说文本生成一个仅包含人名的文件（格式：bookname-r-0000）

4.1.2 任务原理

I 中文分词及 Map 的处理

利用 ansjseg 工具实现中文分词，ansjseg 支持对中文文本迸行分词，并且可以添加用户自定义的词典，这样就可以把出现在金庸小说里的人物的名字找出来。而且在这里我们使用 **DistributeCache** 存储人名列表。

```
输入：文本段落 value
输出：(bookname, names in this value-paragraph of this book)
@Override
Method setup(context):
```

```
        file <- getCacheFiles()
        while (line <- readFile(file)) not null:
                wuxiaNameList.add(line)
                DicLibrary.insert(line)
                // Self-defined dictionary

Method map(key, value, context)
        filename <- context.getFileName()
// get the filename which paragraph belongs to
        termsList <- ToAnalysis.parse(value)
// the names List of paragraph
        for term in termsList:
                if term.isNR and wuxiaNameList.contains(term):
                        result.add(term, '\t')
        if result.length > 0
                context.write(filename, result)
```

ParMapper 类在 setup 函数中获取 **Distributed Cache** 中的人名列表，并在 map 函数中对段落中的人名分割，只有当人名出现在列表中时，才加入结果。

将人名列表传入 Mapper/Reducer 有三种处理方法：

1. 转化成字符串通过 Configuration() 传递，适用于少量数据
2. 通过 Distributed Cache 机制，**适合本实验**的 namelist.txt 传递
3. 存储到 HDFS，在 Map/Reduce 中读取 HDFS，适用于大量数据。

II 多文件输出及 Reduce 处理

```
输入:  filename, names in paragraph
输出:  each book has a outputfile which contains names
Method reduce(key, <Text>values, context)
        for val in values:
                multiOutputs.write(val, "", key)
```

由于需要给每一本小说生成一个包含该小说所有人物的文件，故需要使用 **MultiOutput** 类。本次实验由于会生成多个文件，故使用第一个函数（后续的两个函数需要在 main 函数中制定 **addNamedOutput**）

```
void write(KEYOUT key, VALUEOUT value, String baseOutputPath)

<K, V> void write(String namedOutput, K key, V value)

<K, V> void write(String namedOutput, K key, V value, String baseOutputPath)
```

需要注意的是：**在 setup 函数中进行初始化，在 cleanup 函数中关闭输出。**

4.1.3 运行结果

```
-rw-r--r--   3 2019st17 hadoop_user          0 2019-07-16 12:01 task1_out/_SUCCESS
-rw-r--r--   3 2019st17 hadoop_user      77660 2019-07-16 12:01 task1_out/金庸01飞狐外传-r-00000
-rw-r--r--   3 2019st17 hadoop_user      21109 2019-07-16 12:01 task1_out/金庸02雪山飞狐-r-00000
-rw-r--r--   3 2019st17 hadoop_user      35565 2019-07-16 12:01 task1_out/金庸03连城诀-r-00000
-rw-r--r--   3 2019st17 hadoop_user     192749 2019-07-16 12:01 task1_out/金庸04天龙八部-r-00000
-rw-r--r--   3 2019st17 hadoop_user     163744 2019-07-16 12:01 task1_out/金庸05射雕英雄传-r-00000
-rw-r--r--   3 2019st17 hadoop_user      12350 2019-07-16 12:01 task1_out/金庸06白马啸西风-r-00000
-rw-r--r--   3 2019st17 hadoop_user     224866 2019-07-16 12:01 task1_out/金庸07鹿鼎记-r-00000
-rw-r--r--   3 2019st17 hadoop_user     188130 2019-07-16 12:01 task1_out/金庸08笑傲江湖-r-00000
-rw-r--r--   3 2019st17 hadoop_user     120324 2019-07-16 12:01 task1_out/金庸09书剑恩仇录-r-00000
-rw-r--r--   3 2019st17 hadoop_user     179041 2019-07-16 12:01 task1_out/金庸10神雕侠侣-r-00000
-rw-r--r--   3 2019st17 hadoop_user      60588 2019-07-16 12:01 task1_out/金庸11侠客行-r-00000
-rw-r--r--   3 2019st17 hadoop_user     197688 2019-07-16 12:01 task1_out/金庸12倚天屠龙记-r-00000
-rw-r--r--   3 2019st17 hadoop_user      86037 2019-07-16 12:01 task1_out/金庸13碧血剑-r-00000
-rw-r--r--   3 2019st17 hadoop_user       6346 2019-07-16 12:01 task1_out/金庸14鸳鸯刀-r-00000
-rw-r--r--   3 2019st17 hadoop_user         16 2019-07-16 12:01 task1_out/金庸15越女剑-r-00000
```

图 1: 多文件输出列表

```
程灵素 程灵素 姬晓峰 胡斐 福康安 程灵素
胡斐 程灵素 胡斐 程灵素
胡斐 程灵素 马春花 程灵素 胡斐 马春花 福康安 胡斐 程灵素 程灵素 马春花
徐铮 马行空 脚夫
程灵素
胡斐 姬晓峰
田归农 田归农 田归农
姬晓峰 姬晓峰 胡斐 姬晓峰 胡斐 马春花 马春花 福康安 姬晓峰 胡斐 胡斐
姬晓峰 胡斐 姬晓峰
姬晓峰 胡大哥 姬晓峰
姬晓峰 姬晓峰 胡斐 姬晓峰 胡斐 姬晓峰 姬晓峰
姬晓峰 胡斐 姬晓峰 程灵素 胡斐 姬晓峰 胡斐 胡斐 姬晓峰 胡斐
姬晓峰 姬晓峰 胡斐 姬晓峰 胡斐
```

图 2: 文件内容

图 3: 集群运行结果

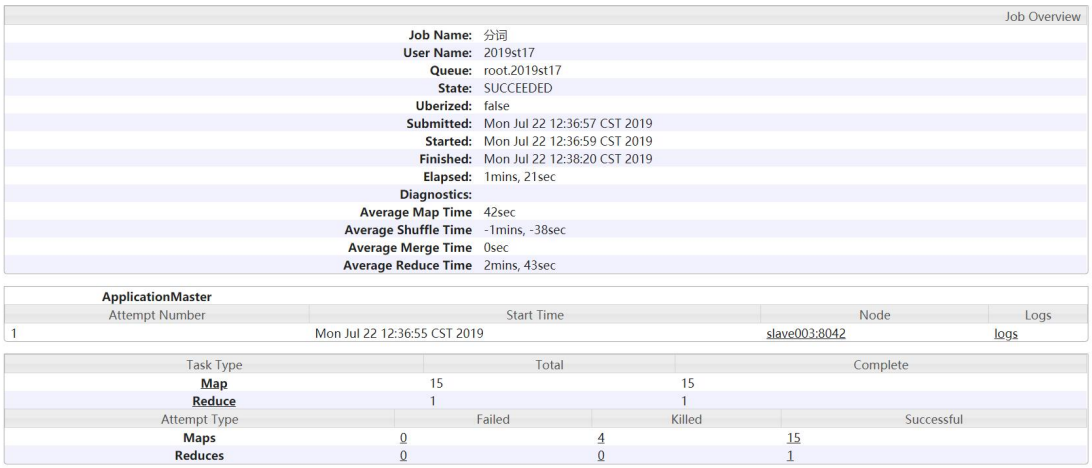


图 4: 集群 History

4.2 二 特征抽取：人物同现统计

本任务的重要完成基于单词同现算法的人物同现统计。在人物同现分析中，如果两个人在原文的同一段落中出现，则认为两个人发生了一次同现关系。我们需要对人物之间的同现关系次数迸行统计，同现关系次数越多，则说明两人的关系越密切。

4.2.1 任务简述

任务 2 分为图 5, 图 6，首先读取任务 1 的输出，去除重复名字后，遍历所有组合，将其数字设为 1；第二步，将所有键值相同的组合累加，输出每个键值出现的总次数。

4.2.2 任务原理

使用 **HashSet** 去除重复人名，再进行人名的两两组合

```
输入：分词后仅保留人名的金庸武侠小说全集（每行表示一个段落）
输出：key: <name1, name2> value: 1
Method Map(value):
        array[] <- value.split('\t')
        for all t in array[]:
                for all u in array[]:
                        emit(<t, v>, one)

输入：key: <name1, name2> value: 1
输出：key: <name1, name2> value: 同现次数
Function Reduce(key, values[])
        for val in values:
                sum += val
        emit(key, sum)
```

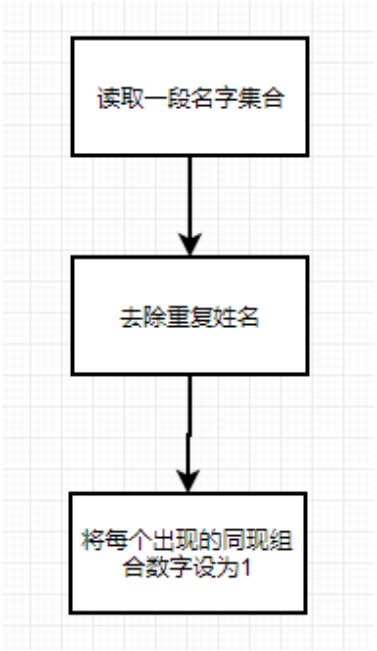


图 5: Map

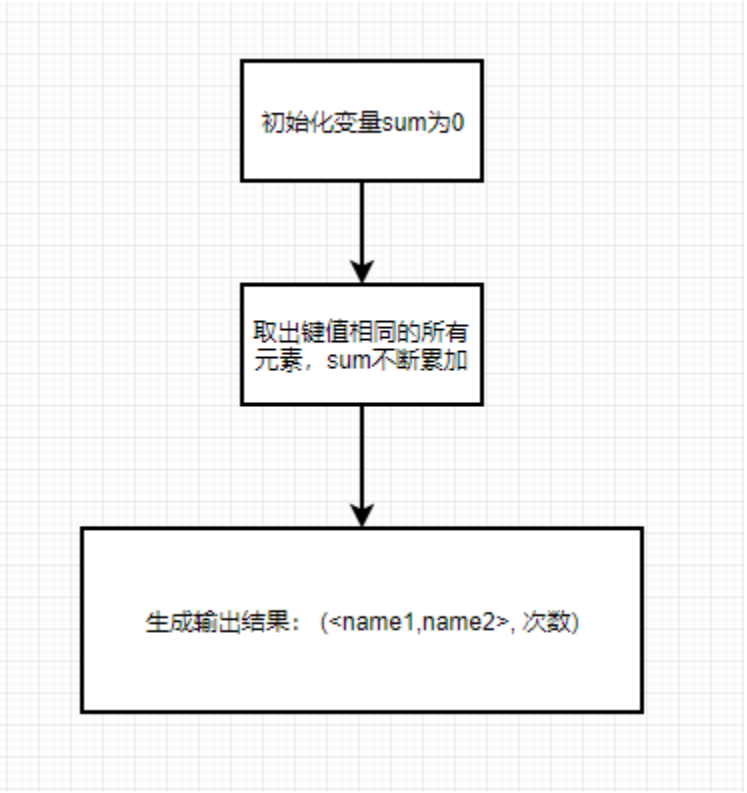


图 6: Reduce

4.2.3 运行结果

<鹿杖客，掌钵龙头> 1
<鹿杖客，摩诃巴思> 1
<鹿杖客，空性> 1
<鹿清笃，申志凡> 1
<鹿清笃，赵志敬> 14
<麦少帮主，殷素素> 2
<黄二毛子，袁承志> 3
<黄伯流，方证> 1
<黄伯流，绿竹翁> 1
<黄国柏，方证> 1
<黄宗羲，马超兴> 1
<黄甫，张勇> 1
<黄甫，韦小宝> 4
<黄真，木桑道长> 1

图 7: 输出文件格式

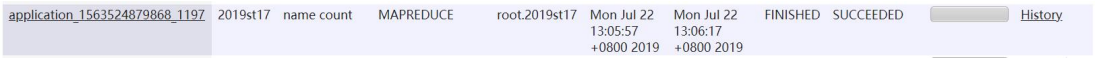


图 8: 集群运行结果

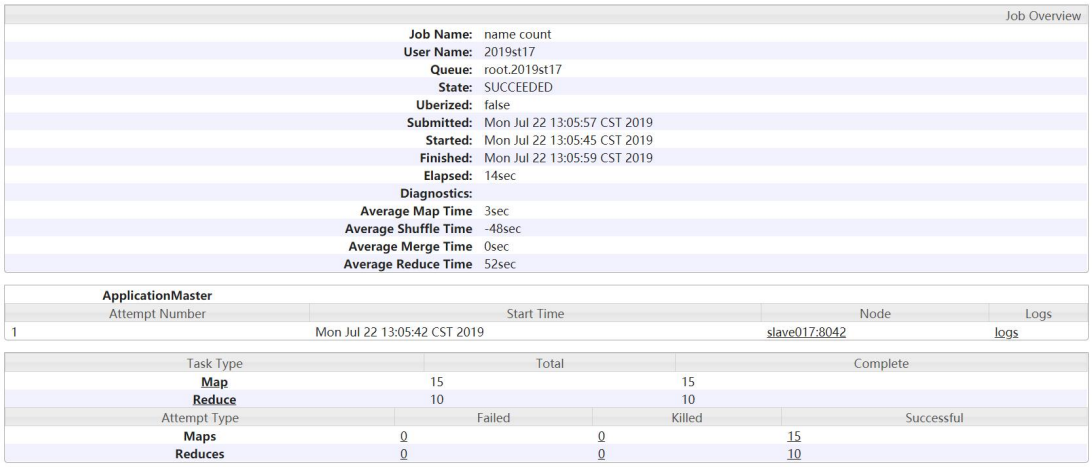


图 9: 集群 History

4.3 三 特征处理：人物关系图构建不特征归一化

4.3.1 任务简述

当获取了人物之间的共现关系之后，我们就可以根据共现关系，生成人物之间的关系图了。人物关系图使用邻接表的形式表示，方便后面的 PageRank 计算。在人物关系图中，人物是顶点，人物之间的互劢关系是边。人物之间的互劢关系靠人物之间的共现关系确定。如果两个人之间具有共现关系，则两个人之间就具有一条边。两人之间的共现次数体现出两人关系的密切程度，反映到共现关系图上就是边的权重。边的权重越高则体现了两个人的关系越密切。

为了使后面的方便分析，还需要对共现次数迚行归一化处理：将共现次数转换为共现概率。

数据输入: 任务二的输出
数据输出: 归一化后的人物关系图

4.3.2 任务原理

```
Mapper
输入: key:<name1,name2> value: 同现次数
输出: key:name1 value:<name2, 同现次数 >
Function map(Text key, Text value):
        name1, name2 <- key
        Emit(name1, <name2,value>)

Reduce
输入: key:name value: 列表, 里面每个元素为 <name2, n>
输出: key:name value: 字符串”name2,n2|name3,n3|…”
Function reduce(Text key, list<Text> values):
        postList <- new 一个字符串列表
        sum <- 0
        for all value in values:
                postList.append(value)
                name2, n <- value
                sum += n
        realValue <- “”
        for all value in postList:
                name2, n <- value
                realValue += (name2 + “,” + n / sum + “|”)
        realValue[last] <- “]”
        realValue <- “[” + realValue
        Emit(name, realValue)
```

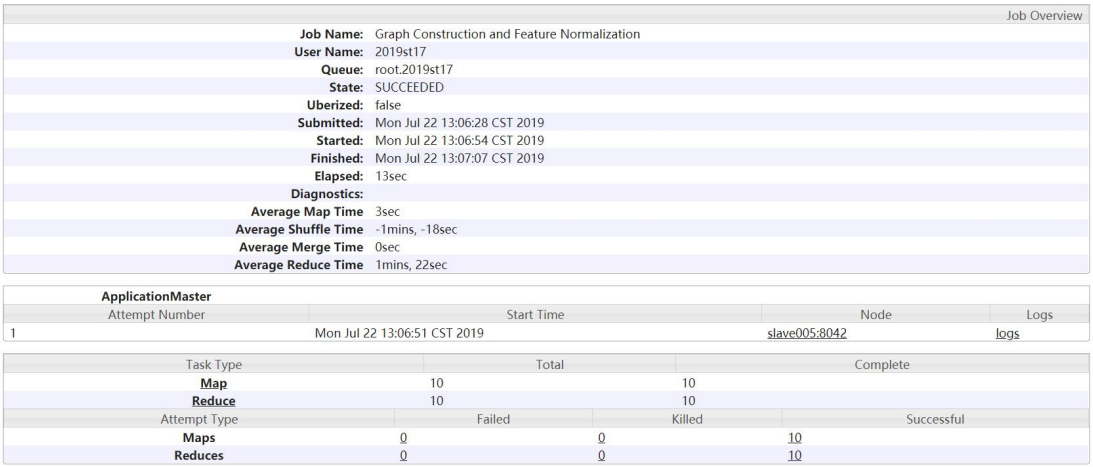
4.3.3 运行结果

丁勉 [岳灵珊,0.011904761904761904|陆伯,0.011904761904761904|桃根仙,0.059523809523809524|英颖,0.011904761904761904|费彬,0.08333333333333333|桃枝仙,0.047619047619047616|胖子,0.023809523809523808|刘正风,0.10714285714285714|方证,0.011904761904761904|桃花仙,0.03571428571428571|米为义,0.011904761904761904|曲洋,0.023809523809523808|定逸,0.023809523809523808|封不平,0.047619047619047616|风清扬,0.011904761904761904|汉子,0.023809523809523808|左冷禅,0.07142857142857142|玉玑子,0.011904761904761904|冲虚道长,0.011904761904761904|大汉,0.011904761904761904|桃叶仙,0.023809523809523808|桃干仙,0.047619047619047616|向大年,0.023809523809523808|岳不群,0.07142857142857142|桃实仙,0.023809523809523808|狄修,0.011904761904761904|莫大,0.011904761904761904|令狐冲,0.07142857142857142|丛不弄,0.03571428571428571|何足道,0.011904761904761904|任我行,0.011904761904761904]
上官铁生 [文酵翁,0.03571428571428571|桑飞红,0.32142857142857145|上官,0.03571428571428571|无常鬼,0.07142857142857142|程灵素,0.07142857142857142|汤沛,0.03571428571428571|福康安,0.10714285714285714|海兰弼,0.03571428571428571|曾铁鸣,0.03571428571428571|郭玉堂,0.07142857142857142|胡斐,0.17857142857142858]

图 10: 输出文件格式

application_1563524879868_1198	2019st17	Graph Construction and Feature Normalization	MAPREDUCE	root.2019st17	Mon Jul 22 13:06:28 +0800 2019	Mon Jul 22 13:06:46 +0800 2019	FINISHED	SUCCEEDED	History

图 11: 集群运行结果



Job Overview

Job Name:	Graph Construction and Feature Normalization
User Name:	2019st17
Queue:	root.2019st17
State:	SUCCEEDED
Uberized:	false
Submitted:	Mon Jul 22 13:06:28 CST 2019
Started:	Mon Jul 22 13:06:54 CST 2019
Finished:	Mon Jul 22 13:07:07 CST 2019
Elapsed:	13sec
Diagnostics:	
Average Map Time	3sec
Average Shuffle Time	-1mins, -18sec
Average Merge Time	0sec
Average Reduce Time	1mins, 22sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Mon Jul 22 13:06:51 CST 2019	slave005:8042	logs

Task Type	Total	Complete
Map	10	10
Reduce	10	10

Attempt Type	Failed	Killed	Successful
Maps	0	0	10
Reduces	0	0	10

图 12: 集群 History

4.4 四 数据分析：基于人物关系图的 PageRank 计算

4.4.1 任务简介

在这一部分，需要实现的功能是：对任务关系图进行一个数据分析，通过 PageRank，我们就可以定量地分析金庸武侠江湖中的“主角”们是哪些。PageRank 任务总共分为三个部分，首先将获取的数据格式化为能够处理的格式，其次进行迭代运算，最后将结果进行整理。

共分为 4 个类,**CorrectFormat.class, PageRanking.class, SortPageRank.class, PageRankDriver.class**

4.4.2 实验原理

I Driver 主函数

函数调用顺序如图 13：前一轮的输出为后一轮的输入，PageRanking 会迭代执行，直到满足条件：（人物排名不再改变或迭代固定次数，在这里我们使用固定次数 **TIMES**）

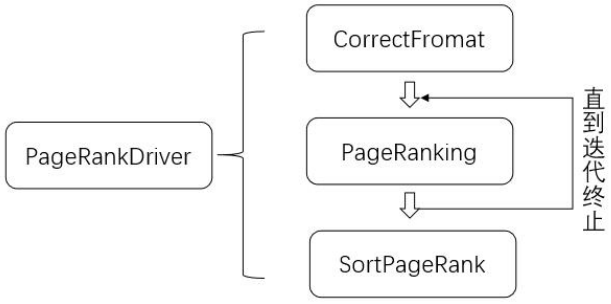


图 13: 主函数框架

II 调整数据格式

任务 3 的输出格式如图 14; 经过格式修改后得到的格式如图 15（其中 PR_INIT 即为每个人物 PR 的初始值）。

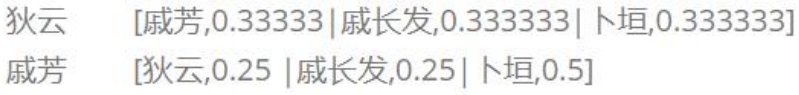


图 14: 格式修改前

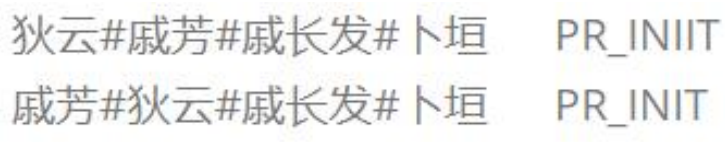


图 15: 格式修改后

Map 函数的处理 (key, value 的选取)

```
输入:  name [name,heavy|name,heavy]
输出:  (name#name#name PR\_INIT)
Method Map(Text value, Context context):
        leaderName <- value.toString.split(' ')[0]
        followItems <- value.toString.split(' ')[1]
        // get list (name, heavy)
        String[] followName <- followItems.split('|' or '[')
        StringBuilder result <- new StringBuilder();
        for items in followName[]:
                name <- items.split(',')[0]
                if name.length > 0:
                        result.append('#' + name)
        context.write(Text(result), PR_INIT)
        // name_link list and PR value
```

Reduce 函数按照接收的 key, value 写入文件即可.

III 迭代 PageRank

由于需要进行迭代，故每次写入的文件格式都与 **4.2** 的输出格式相同。

Map 函数的处理:

```

输入:  name#name#name valuePR
输出:  (name+valuePR one) OR  (name#name#name one)
Method Map(Text value, Context context):
        names <- value.toString.split('\t')[0]
        // get name_link_list
        pr <- value.toString.split('\t')[1]
        // get pr value
        items[] <- names.split('#')
        leaderPR <- pr.value / len(items.length-1)
        for i = 1 to items.length:
                temp <- items[i] + leaderPR
                context.write(Text(temp), 1)
        context.write(Text(names), 1)
```

在最后仍然需要给出人物的链接关系，以免丢失信息，便于下一次的迭代

NewPartitioner 自定义 Partition 函数，将键值对分组到 Reduce 中。

```

```

```
输入: (name+valuePR one) OR (name#name#name one)
操作: 找出两者的第一个 name, 并作为参数调用父类的 getPartition 函数
输出: (name+valuePR one) OR (name#name#name one)
@Override
Method int getPartition(key, value)
        if key.toString.contains('+'):
                term <- key.toString.split('+')[0]
        else
                term <- key.toString.split('#')[0]
        return super.Partition(term, value)
```

如果 key 中包含'+', 则表明是 name+pr 的结构，否则即为 name_link_list 的结构；需要保证的是：后者的出发节点和前者的人物在同一个 Reduce 任务中

Reduce 函数的处理:

```
输入: (name+valuePR one) OR (name#name#name one)
输出: (name#name#name  valuePR)
Method Reduce(key, values, context)
        //get leaderName as NewPartition's term
        leaderName <- key.getLeaderName
        initialize cur <- leaderName
        // just do it once
        if not cur.equals(leaderName):
                cleanup(context)
        // call the cleanup to write info
        if key.toString.contains('+')
        // key format: name+pr
                for v in values[]:
                        valuePR += (PR in key)
        // the same name+pr may accur several times
        else:
                linkList <- key.toString()
        // key format: name#name#name

@Override
Method cleanup(context)
        valuePR <- (1-PR\_CHANGE)+PR\_CHANGE*valuePR
        context.write(linkList, valuePR)
        // emit format: name#name#name  valuePR
        valuePR <- 0
```

重载 cleanup 函数，当发现接收到的 key 值中的 leaderName 发生变化时，表明：上一个 leaderName 的 PR 已经计算完成，需要将其保存，保存的格式为：name_link_list ''^ valuePR

IV 数据排序

1. 在 Map 函数中以每个人物的 pr 值作为 key, 人名为 value 发送出去，经过默认的 partition 函数发给 Reduce 任务（在这里，我们设置 Reduce 任务的个数为 1）

2. 在 Reduce 函数中重写 DoubleWritable.Comparator 类的 comapre 函数。

```
@Override
Method compare(byte[] b1, int s1, int l1,
                    byte[] b2, int s2, int l2)
    return -super.compare(b1, s1, l1, b2, s2, l2);
```

3. 最后，在 Reduce 函数中按照 key, value 的顺序写入文件即可。

4.4.3 运行结果

```
汉子    21.54764055814416
大汉    11.234918457852404
韦小宝  9.463820469248043
胡斐    6.4512169931556915
张无忌  6.202936715794956
郭靖    5.603366108609243
吴三桂  5.594635633383974
令狐冲  5.505702117697552
说不得  5.404924754099813
段誉    5.34009665617458
袁承志  4.841691633021963
哑巴    4.839887488282016
莫大    4.768532283429151
老头子  4.672837171068772
黄蓉    4.4919795485182945
杨过    4.395339480893321
```

图 16: 输出文件格式

application_1563524879868_1213	2019st17	SortPageRank	MAPREDUCE	root.2019st17	Mon Jul 22 13:17:50 +0800 2019	Mon Jul 22 13:18:06 +0800 2019	FINISHED	SUCCEEDED	History
application_1563524879868_1212	2019st17	PageRanking	MAPREDUCE	root.2019st17	Mon Jul 22 13:17:28 +0800 2019	Mon Jul 22 13:17:47 +0800 2019	FINISHED	SUCCEEDED	History
application_1563524879868_1211	2019st17	PageRanking	MAPREDUCE	root.2019st17	Mon Jul 22 13:17:06 +0800 2019	Mon Jul 22 13:17:26 +0800 2019	FINISHED	SUCCEEDED	History
application_1563524879868_1210	2019st17	PageRanking	MAPREDUCE	root.2019st17	Mon Jul 22 13:16:45 +0800 2019	Mon Jul 22 13:17:04 +0800 2019	FINISHED	SUCCEEDED	History
application_1563524879868_1209	2019st17	PageRanking	MAPREDUCE	root.2019st17	Mon Jul 22 13:16:24 +0800 2019	Mon Jul 22 13:16:43 +0800 2019	FINISHED	SUCCEEDED	History
application_1563524879868_1208	2019st17	PageRanking	MAPREDUCE	root.2019st17	Mon Jul 22 13:16:02 +0800 2019	Mon Jul 22 13:16:22 +0800 2019	FINISHED	SUCCEEDED	History
application_1563524879868_1207	2019st17	CorrectFormat	MAPREDUCE	root.2019st17	Mon Jul 22	Mon Jul 22	FINISHED	SUCCEEDED	History

图 17: 集群运行结果

Job Overview

Job Name:	CorrectFormat
User Name:	2019st17
Queue:	root.2019st17
State:	SUCCEEDED
Uberized:	false
Submitted:	Mon Jul 22 13:15:43 CST 2019
Started:	Mon Jul 22 13:13:01 CST 2019
Finished:	Mon Jul 22 13:13:13 CST 2019
Elapsed:	11sec
Diagnostics:	
Average Map Time	3sec
Average Shuffle Time	1mins, 41sec
Average Merge Time	0sec
Average Reduce Time	-1mins, -37sec

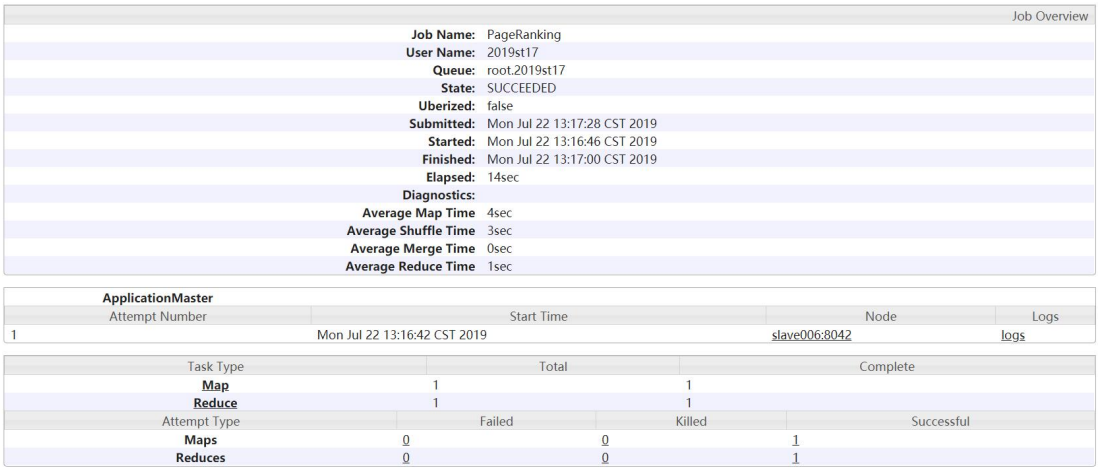
ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Mon Jul 22 13:12:58 CST 2019	slave008:8042	logs

Task Type	Total	Complete
Map	10	10
Reduce	1	1

Attempt Type	Failed	Killed	Successful
Maps	0	0	10
Reduces	0	0	1

图 18: 集群 **CorrectFormat** History



Job Overview

Job Name:	PageRanking
User Name:	2019st17
Queue:	root.2019st17
State:	SUCCEEDED
Uberized:	false
Submitted:	Mon Jul 22 13:17:28 CST 2019
Started:	Mon Jul 22 13:16:46 CST 2019
Finished:	Mon Jul 22 13:17:00 CST 2019
Elapsed:	14sec
Diagnostics:	
Average Map Time	4sec
Average Shuffle Time	3sec
Average Merge Time	0sec
Average Reduce Time	1sec

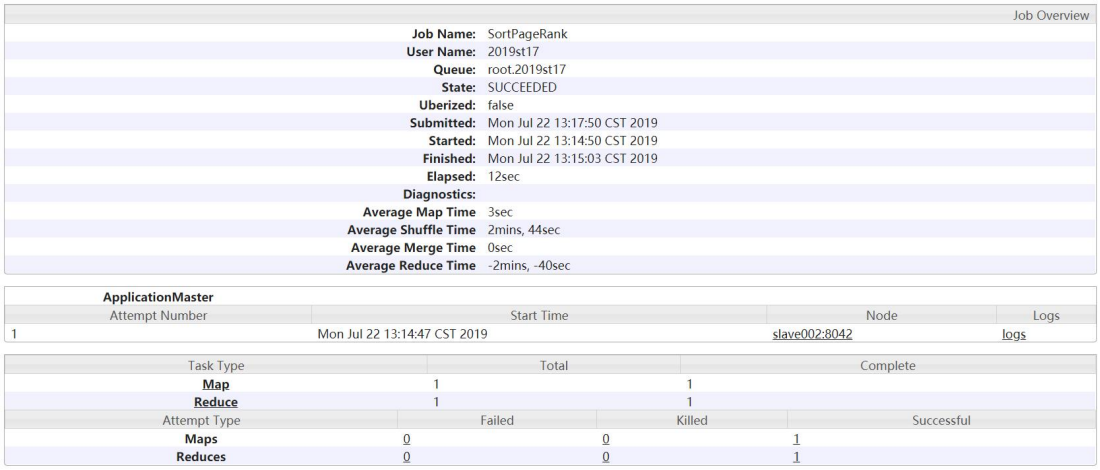
ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Mon Jul 22 13:16:42 CST 2019	slave006:8042	logs

Task Type	Total	Complete
Map	1	1
Reduce	1	1

Attempt Type	Failed	Killed	Successful
Maps	0	0	1
Reduces	0	0	1

图 19: 集群 **PageRank**History



Job Overview

Job Name:	SortPageRank
User Name:	2019st17
Queue:	root.2019st17
State:	SUCCEEDED
Uberized:	false
Submitted:	Mon Jul 22 13:17:50 CST 2019
Started:	Mon Jul 22 13:14:50 CST 2019
Finished:	Mon Jul 22 13:15:03 CST 2019
Elapsed:	12sec
Diagnostics:	
Average Map Time	3sec
Average Shuffle Time	2mins, 44sec
Average Merge Time	0sec
Average Reduce Time	-2mins, -40sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Mon Jul 22 13:14:47 CST 2019	slave002:8042	logs

Task Type	Total	Complete
Map	1	1
Reduce	1	1

Attempt Type	Failed	Killed	Successful
Maps	0	0	1
Reduces	0	0	1

图 20: 集群 **SortPageRank**History

4.5 五 数据分析：人物关系图的标签传播

4.5.1 任务简述

标签传播（LabelPropagation）是一种半监督的图分析算法，他能为图上的顶点打标签，迚行图顶点的聚类分析，从而在一张类似社交网络图中完成社区发现（Community Detection）。将金庸小说中的人物完成社区发现。

输入：任务 3 的输出，人物之间的连接关系

输出：人物及其标签

4.5.2 任务原理

标签传播可以分成以下三步骤：

I 数据标准化

这与任务三有点像：不同点在于：未归一化并且人物都会有一个标签。格式为：

name, label Link

其中 Link 为其相邻点及其共现次数。name1,n1|name2,n2|name3,n3 以 | 为分隔符

II 标签传播迭代

```
1 Mapper
2 输入: key: name, label value: name1,n1|name2,n2|name3,n3
```

```
输出: key: name(i) value:<name,label,ni>
Function map(Text key, Text value):
        name, label <- key
        for all one in value:
                name(i), n(i) <- one
                Emit(name(i), <name,label,n(i)>)
Reducer
输入: key: name value: 以 <name, label, ni> 格式的列表
输出: key name, label value: name1,n1|name2,n2|name3,n3
Function reduce(Text text, list<Text> values):
        map <- new Map<String, int>()
        realValue = ""
        for all value in values:
                name,label,n <- value
                if(label in map.keys)
                        map(label) = 0
                else
                        map(label) += n
                realValue += (name+','+n+'|')
                label <- map 中的值的最大的 key
                Emit(<name,label>, realValue)
```

III 统计标签

当迭代收敛或者到达一定次数停止

将最后一次的结果转化成 label, name 的形式即可。

4.5.3 运行结果

如图 21，表示每个人物所属的人名，可以转换成便于查看的结果，如图 22。

```
陈家洛 张召重
陈家洛 心砚
陈家洛 常赫志
陈家洛 骆冰
陈家洛 于万亭
陈家洛 陈正德
陈家洛 马大挺
陈家洛 玉如意
陈家洛 德布
陶子安 静智大师
陶子安 周云阳
陶子安 田青文
陶子安 陶子安
韦小宝 吴大鹏
韦小宝 多隆
韦小宝 张妈
韦小宝 徐天川
韦小宝 昌齐
韦小宝 明珠
韦小宝 杜立德
韦小宝 海大富
韦小宝 王进宝
韦小宝 章老三
韦小宝 米思翰
```

图 21: 标签 & 人名

```
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/丁大全
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/乔峰
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/令狐冲
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/卓天雄
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/司马林
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/张无忌
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/曹云奇
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/本因
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/杨过
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/段誉
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/狄云
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/盖一鸣
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/石破天
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/耶律涅鲁古
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/耶律重元
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/胡斐
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/花剑影
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/苏普
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/袁承志
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/郑板桥
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/郭靖
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/陈大方
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/陈家洛
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/陶子安
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/韦小宝
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/黄蓉
```

图 22: 有相同标签的写入同一个文件

application_1563524879868_1206	2019st17	LPA Rank	MAPREDUCE	root.2019st17	Mon Jul 22 13:09:38 +0800 2019	Mon Jul 22 13:09:56 +0800 2019	FINISHED	SUCCEEDED	History
application_1563524879868_1205	2019st17	LPAltr	MAPREDUCE	root.2019st17	Mon Jul 22 13:09:17 +0800 2019	Mon Jul 22 13:09:36 +0800 2019	FINISHED	SUCCEEDED	History
application_1563524879868_1204	2019st17	LPAltr	MAPREDUCE	root.2019st17	Mon Jul 22 13:08:57 +0800 2019	Mon Jul 22 13:09:15 +0800 2019	FINISHED	SUCCEEDED	History
application_1563524879868_1203	2019st17	LPAltr	MAPREDUCE	root.2019st17	Mon Jul 22 13:08:37 +0800 2019	Mon Jul 22 13:08:55 +0800 2019	FINISHED	SUCCEEDED	History
application_1563524879868_1202	2019st17	LPAltr	MAPREDUCE	root.2019st17	Mon Jul 22 13:08:17 +0800 2019	Mon Jul 22 13:08:35 +0800 2019	FINISHED	SUCCEEDED	History
application_1563524879868_1201	2019st17	LPAltr	MAPREDUCE	root.2019st17	Mon Jul 22 13:07:57 +0800 2019	Mon Jul 22 13:08:15 +0800 2019	FINISHED	SUCCEEDED	History
application_1563524879868_1200	2019st17	LPAltr	MAPREDUCE	root.2019st17	Mon Jul 22 13:07:36 +0800 2019	Mon Jul 22 13:07:55 +0800 2019	FINISHED	SUCCEEDED	History

图 23: 集群运行结果

Job Overview

Job Name:	LPAltr
User Name:	2019st17
Queue:	root.2019st17
State:	SUCCEEDED
Uberized:	false
Submitted:	Mon Jul 22 13:07:36 CST 2019
Started:	Mon Jul 22 13:04:54 CST 2019
Finished:	Mon Jul 22 13:05:07 CST 2019
Elapsed:	13sec
Diagnostics:	
Average Map Time	3sec
Average Shuffle Time	1mins, 37sec
Average Merge Time	0sec
Average Reduce Time	-1mins, -33sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Mon Jul 22 13:04:51 CST 2019	slave008:8042	logs

Task Type	Total	Complete
Map	10	10
Reduce	10	10

Attempt Type	Failed	Killed	Successful
Maps	0	0	10
Reduces	0	0	10

图 24: 集群 **LPAltr**History

Job Overview

Job Name:	LPA Rank
User Name:	2019st17
Queue:	root.2019st17
State:	SUCCEEDED
Uberized:	false
Submitted:	Mon Jul 22 13:09:38 CST 2019
Started:	Mon Jul 22 13:06:55 CST 2019
Finished:	Mon Jul 22 13:07:08 CST 2019
Elapsed:	13sec
Diagnostics:	
Average Map Time	3sec
Average Shuffle Time	1mins, 41sec
Average Merge Time	0sec
Average Reduce Time	-1mins, -36sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Mon Jul 22 13:06:52 CST 2019	slave008:8042	logs

Task Type	Total	Complete
Map	10	10
Reduce	1	1

Attempt Type	Failed	Killed	Successful
Maps	0	0	10
Reduces	0	0	1

图 25: 集群 **LPA Rank**History

4.5.4 数据可视化

使用软件 **Gephi**, 由给出的标签人物信息，自动生成图，更直观展示人物之间的关系。

通过如图 26指令获取人物顶点和边的关系，points.csv & edges.csv，导入软件即可获取如图 27和图 28, 局部放大图 29的效果

```
sed 's/\t/,/g; s/,\(.*\)/,\1,\1/g; 1i\class,id,label' points > points.csv
sed 's/[<>]*//g; s/\t/,/g; 1i\Source,Target,Weight' edges > edges.csv
```

图 26: 获取边及顶点

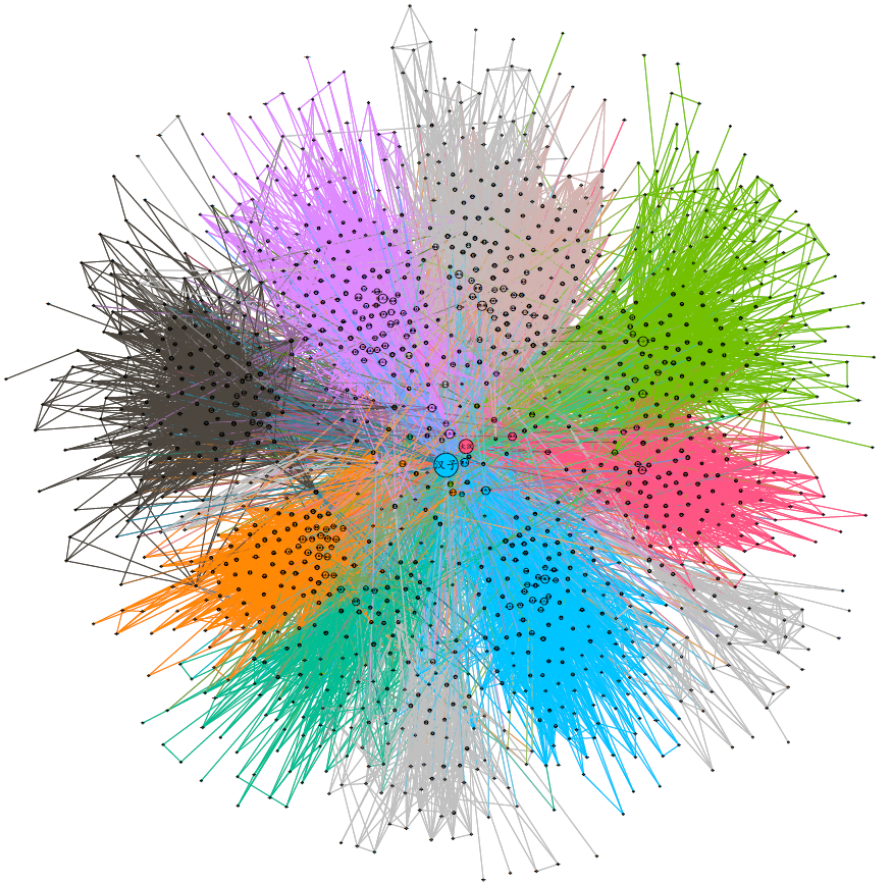


图 27: 人物关系图

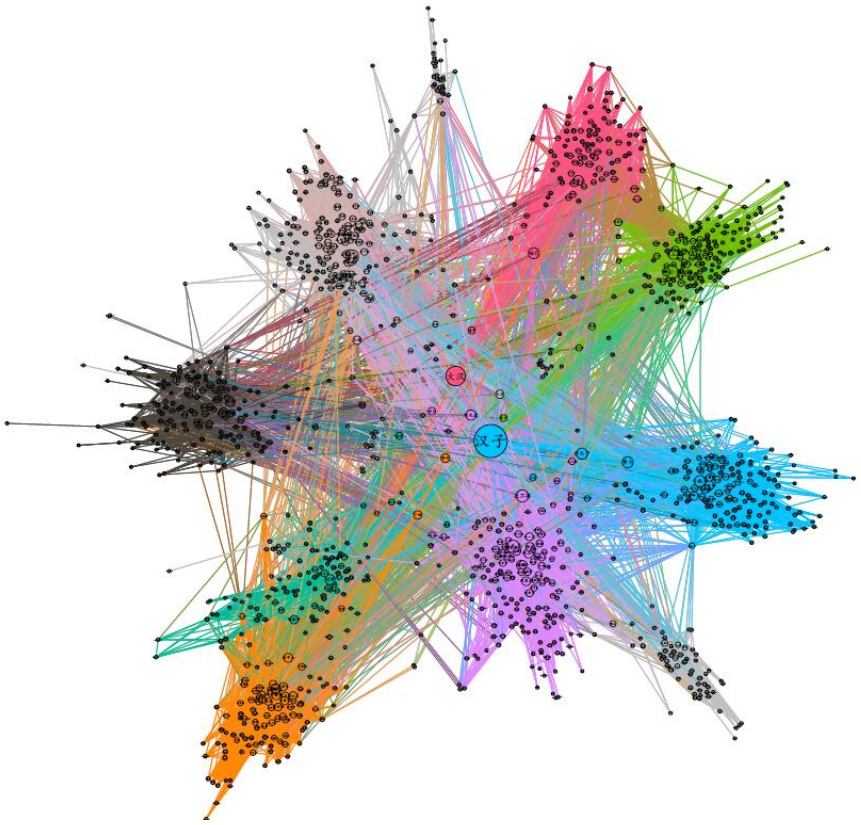


图 28: 人物关系图 2

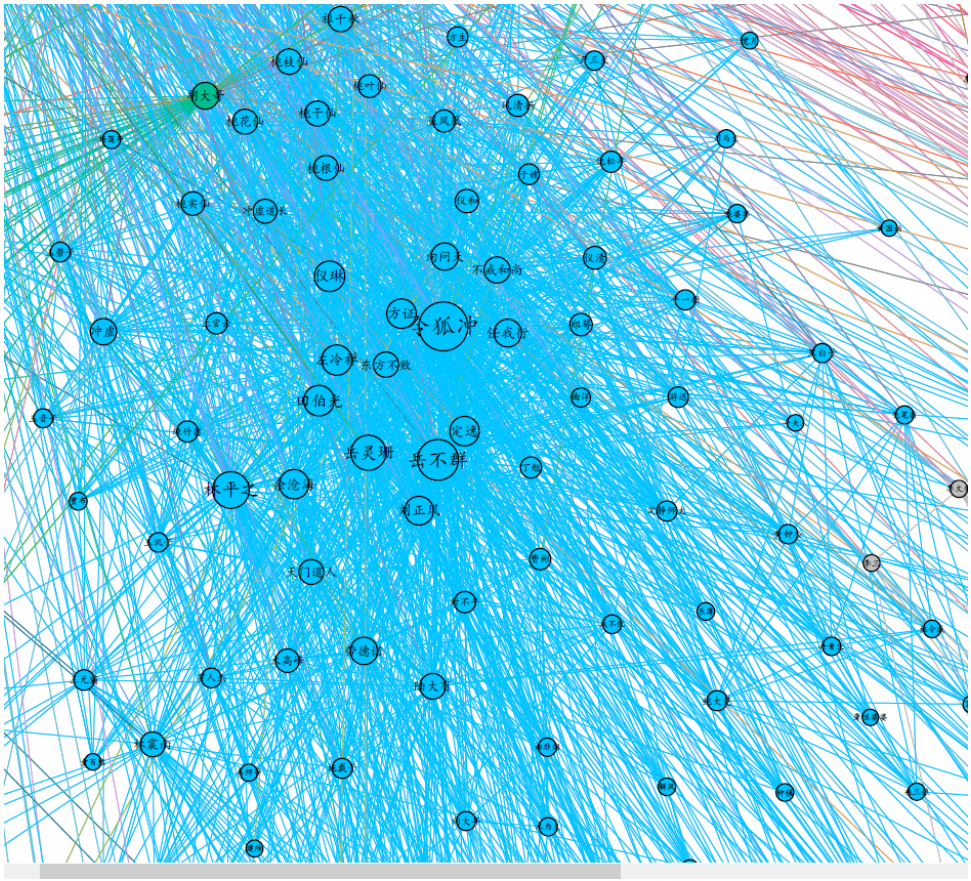


图 29: 局部放大《笑傲江湖》

4.6 六 分析结果整理

4.6.1 任务简述

任务 4 和任务 5 默认的输出内容是杂乱的，从中无法直接的得到分析结论，因此我们需要对上述分析的结果迸行整理、从而使人可以很容易地从分析结果中发现一些有趣的结论。

4.6.2 运行结果

由于该任务涉及的 **MultiOutput** 多文件输出，定制 **Partition** 等原理在之前的任务中已经介绍过，故不再赘述

I 任务 4 输出，排序

```
汉子    21.54764055814416
大汉    11.234918457852404
韦小宝  9.463820469248043
胡斐    6.4512169931556915
张无忌  6.202936715794956
郭靖    5.603366108609243
吴三桂  5.594635633383974
令狐冲  5.505702117697552
说不得  5.404924754099813
段誉    5.34009665617458
袁承志  4.841691633021963
哑巴    4.839887488282016
莫大    4.768532283429151
老头子  4.672837171068772
黄蓉    4.4919795485182945
杨过    4.395339480893321
```

图 30: 按照 PR 从大到小排序

II 任务 6 输出，将属于同一个标签的任务输出到一起，便于查看标签传播的结果

```
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/丁大全
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/乔峰
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/令狐冲
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/卓天雄
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/司马林
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/张无忌
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/曹云奇
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/本因
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/杨过
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/段誉
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/狄云
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/盖一鸣
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/石破天
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/耶律涅鲁古
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/耶律重元
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/胡斐
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/花剑影
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/苏普
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/袁承志
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/郑板桥
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/郭靖
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/陈大方
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/陈家洛
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/陶子安
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/韦小宝
drwxr-xr-x   - 2019st17 hadoop_user          0 2019-07-16 12:15 LPA/result/黄蓉
```

图 31: 属于同一个标签的在同一个文件中

5 实验心得体会

I 通过参与小组形式的实验，体验到成员合作的重要性，每个人都有分工，一起商讨问题的解决方案

II 善于从书本、网络上寻找 Bug 的解决方案，耐心调试

III 完成了一个使用 mapreduce 框架进行数据分析的流程，了解了基本步骤

6 致谢

感谢《大数据》这门课程的顾荣老师和助教们的答疑解惑，同时也感谢与我们组进行探讨交流的其他小组。最后谢谢小组各位成员的努力和付出，我们得以成功完成本门课程的大实验。