**大数据处理期末课程设计**

系别：计算机科学与技术

组号：2019st07

组员：161220106 施雨杰，161220011 陈嘉言

161220012 陈剑豪，161220123 王荆涛

|  |  |
| --- | --- |
| **实验选题** | ——“金庸的江湖” |
| **主要内容** | 1. 借助Ansj\_Seg工具的中文分词与人名提取。 2. 基于MapReduce计算框架的人物关系分析。（具体包含关系权值归一化，人物PageRank值计算，标签传播……） 3. 基于Spark的对上述功能的复现。 4. 借助Gephi工具的结果图示化呈现。 5. 运用flask与bootstrap框架搭建搜索引擎实现金庸的江湖问答系统 |

|  |  |  |  |
| --- | --- | --- | --- |
| 计算模型 | **MapReduce**（必做部分） | **Spark**（选做部分） | **Search Engine**(额外内容) |
| 基础架构 | hadoop（版本：2.7.1）  https://ss1.baidu.com/6ONXsjip0QIZ8tyhnq/it/u=1523529781,231372822&fm=58&bpow=563&bpoh=447 | spark（版本：2.4.3） | Flask+Boostrap |
| 编程语言 | Java | Python | Python+html+  CSS+Javascript |
| 开发环境（IDE） | IntelliJ IDEA | Pycharm | Pycharm+Webstorm |

|  |  |
| --- | --- |
| 组内分工 |  |

**备注格式：**

实验中用到的指令用黄色背景标出

截图中的重要信息用红色椭圆或方框标出

**目录**

1. [说明](#〇、说明)

一、基于MapReduce计算框架的数据处理

task1：[中文分词与人名提取](#task1：中文分词与人名提取)

task2：[人物共现关系及次数统计](#task2：人物共现关系及次数统计)

task3：关系权值归一化

task4：人物PageRank值计算

task5：人物关系图标签传播（Label Passing）

task6：任务4、5结果的排序输出

task7：（拓展）任务5所得标签与实际书名的自动化匹配

……

（提取人物出场段落&标签传播正确率检验）

（代码整合与集群运行，附效果截图）】

二、基于Spark计算框架的复现

（1）实现思路简述

（2）性能优化分析

三、数据处理结果的图示化呈现

（1）Gephi工具的安装

（2）数据文件的生成及导入

（3）关系图效果展示（附模型生成视频？简单分析）

四、………

**〇、说明**

本文系“大数据处理综合实验”——期末课程设计实验报告，任务选题为“金庸的江湖”。主要内容是：以金庸十五本武侠小说文本为原始数据，从中提取中指定的人物名，建立人物之间的共现关系，通过计算PR值、计算标签传播等方式发掘其中隐含的信息，并最终通过图示化工具予以清晰呈现。

上述整个数据处理流程都已通过MapReduce和Spark两种计算框架分别实现。由于MapReduce为本课程的主要内容，实验报告的重点也将放在这一部分（具体参见第二章）。Spark实现思路只简单介绍，但会对其相对于MP的性能优化进行简要分析（参见第三章）。

在分任务描述每个MapReduce的task具体实现时，尽量遵循：任务目的 + 输入输出样例 + 算法描述（Setup+Map+Reduce，伪代码or代码） + 实际运行结果（截图）

的格式，尽可能详细的说明每个子任务的实现思路。

在完成课程实验必做内容（共现关系提取、PageRank计算…）和明确选做内容（标签传播、spark复现…）基础上，本小组另外增加了一些相关的拓展内容：…………

最后，由于本实验报告具有备忘的性质，在一些具体操作上描述非常细致，这些内容对于审阅者而言可能是冗长且多余的。本文中会在合适的地方设置跳转链接，帮助审阅者跳过琐碎的操作细节，直接查看最终呈现结果。

[↑返回目录](#Content)

**一、基于MapReduce计算框架的数据处理**

**task1：中文分词与人名提取**

**（1）任务目的：**

借助Ansj\_Seg分词工具，基于给定的namelist.txt人名列表文件，建立关于金庸小说人物名的词库。以15本金庸小说作为输入，进行中文分词，提取出其中指定的人物名。

**（2）输入输出样例：**

（Input：小说中的一段文本）

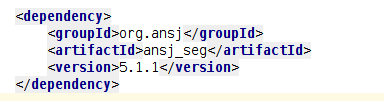
王处一道：“你六位师父既然尚未明言，我也不便代说。”他曾听丘处机说起过前后的原委，对江南六怪的义举心下好生相敬。他和马钰是一般的心思，也盼江南六怪获胜，不过他是师弟，却不便明劝丘师哥相让，今日见了郭靖的为人，暗自思量如何助他一臂之力，却又不能挫折丘师哥的威名，决意届时赶到嘉兴，相机行事，从中调处。

（Output：其中出现的（明确在词库中出现过的）人物名）

王处一 丘处机 马钰 郭靖

**（3）相关配置：**

本实验的MapReduce部分统一使用Intellij IDEA下的Maven项目管理。在项目的pom.xml配置文件<dependencies>……</dependencies>中添加如下依赖：



其中版本一栏选择“5.1.1”。

点击“Import Changes”自动完成配置。



在java源文件开头处添加以下几条依赖：

import org.ansj.domain.Result;

import org.ansj.domain.Term;

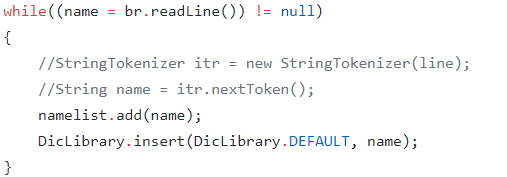
import org.ansj.library.DicLibrary;

import org.ansj.splitWord.analysis.ToAnalysis;

**（4）算法描述：**

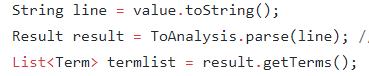
Setup()阶段：

建立人名词库。从助教提供的namelist.txt文件中逐行读取人物名（name），调用ansj库中的“DicLibrary”功能，将读出的name加入词库。同时把name加入一个全局列表namelist，供之后分词时过滤筛选。

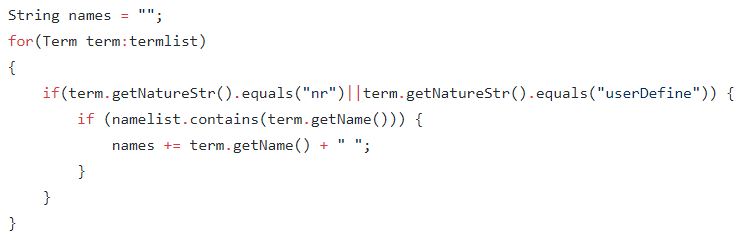


Map()阶段：

对输入的每一行文本，调用ansj库的“ToAnalysis”功能，完成中文分词，并将结果暂存入名为termlist的列表。



遍历termlist列表，筛选出其中可能是人物名的词（词性为“nr”或“userDefine”）。如果这些词确定在之前建立的namelist列表中出现过，则将其写入字符串names。



词库自行识别的人名

setup阶段添加的人名

（\*\*\*这里之所以要重复检查每个term是否在namelist列表中，是因为某些被词库自行识别出的“人名”可能并没有在namelist.txt中出现过。仅凭词性不能完全确定选出的人名是否应该加入输出，必须通过namelist列表进行额外的一轮过滤。）

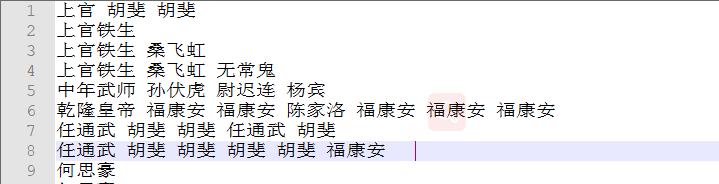
最后，将所得的（非空的）names文本发出即可。

Reduce()阶段：

task1的reduce阶段不作任何处理。直接将从mapper端传来的names文本写入输出文件。

**（5）运行结果**

这是小说《飞狐外传》分词提取结果的开头部分。



**task2：人物共现关系及次数统计**

…… ……

**task6：****任务4、5结果的排序输出**