大规模分布式系统第三次实验报告
员司雨 17307110448

设计 MR 程序，实现具有 payload 的倒排文档索引

一. 实验介绍

mrjob 是编写能够在 hadoop 上运行的 python 程序最简单的途径。如果使用 mrjob，可以在本地测试的代码，甚至不需要安装 hadoop 或者在选择的集群上运行。

另外，mrjob 可以和亚马逊的 EMR（Elastic MapReduce）服务无缝集成。只要设置完毕，就可以运行在 EMR 上，像在自己的笔记本上运行一样简单。

二. 实验环境

1. Ubuntu18.04
2. jdk 1.8.0_131
3. hadoop 2.7.7
4. Python3.6.2
5. mrjob 包

三. 实验过程

PartI:数据预处理

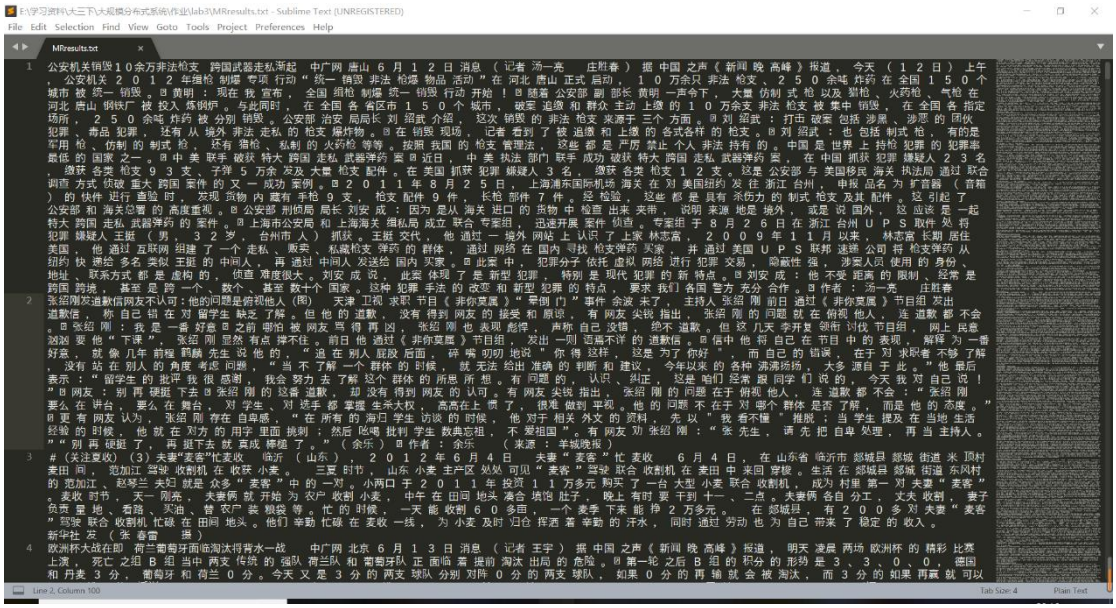
1. 将原文档用 jieba 库分词后预处理为<网页名称 内容>的格式文档。预处理代码如下：

```
1.  import jieba
2.  def write_file(r):
3.      f = open("D:/MRresults.txt", "w", encoding = 'utf-8')
4.
5.      for key in r.keys():
6.          f.write(key + '\t' + r[key])
7.          f.write('\n')
8.      f.close()
9.
10. if __name__ == '__main__':
11.     r = {}
12.     file = open('E:/学习资料/大三下/大规模分布式系统/作业
    /lab3/news_tensite_xml.smarty.txt', 'r', encoding='utf-8')
13.     ls = []
14.     for line in file.readlines():
15.         ls.append(line)
16.
17.
```

```
    l = len(ls)
    for i in range(l):
      if "contenttitle" in ls[i]:
        title = ls[i].replace('<contenttitle>','')
        title = title.replace('</contenttitle>\n','')
        text = ls[i+1].replace('<content>','')
        text = text.replace('</content>\n','')
        text = jieba.cut(text,cut_all=False)
        r[title] = ' '.join(text)

    write_file(r)
```

参见：data_process.py

2. 输出结果



参加：MRresults.txt

PartII：倒排文档生成函数

1. Python 版本查看

```
ics@ubuntu:/usr/bin$ python -V
Python 3.6.2
ics@ubuntu:/usr/bin$ python
Python 3.6.2 (default, Mar 27 2020, 02:59:06)
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("hello world")
hello world
>>> exit()
```

Python 版本为 3.6.2

2. Python 代码编写

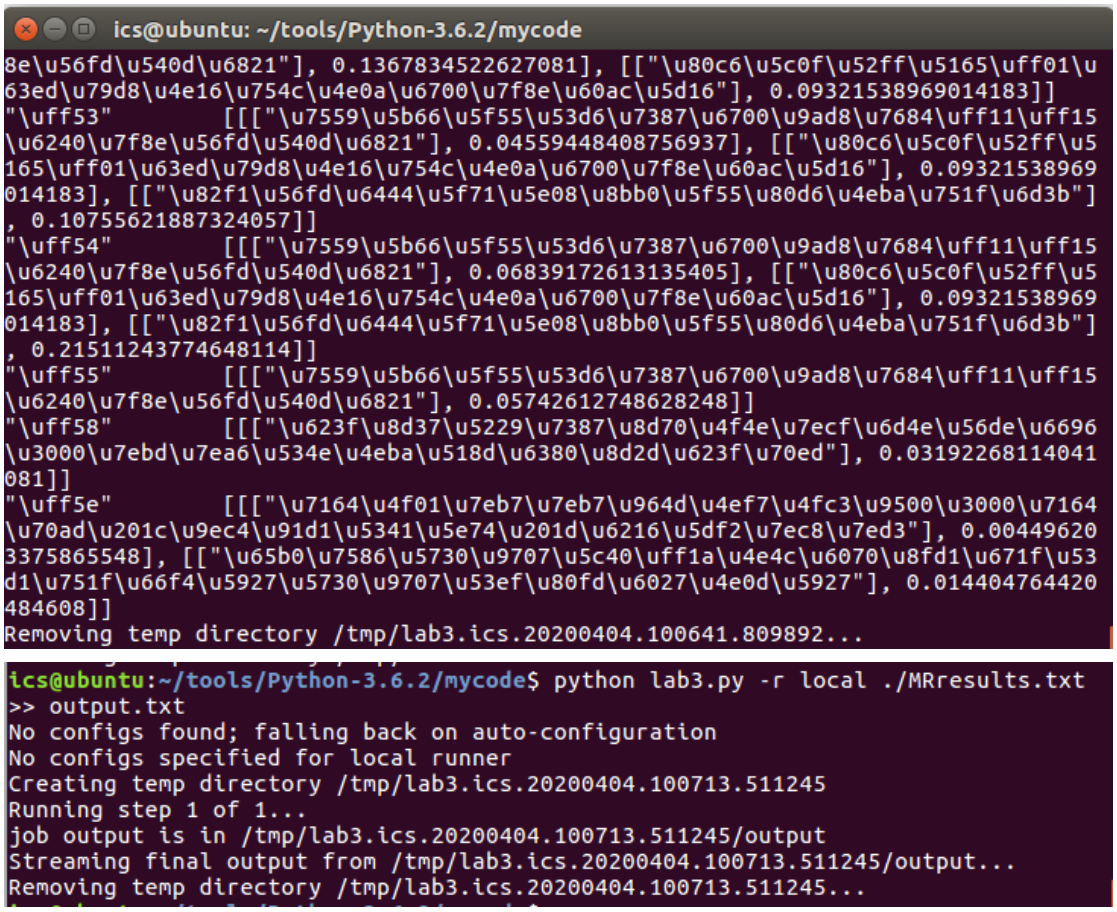
```
1.  from mrjob.job import MRJob
```

```
from mrjob.step import MRStep
import math
import os
from collections import Counter

class MRFILE_TYPE_Counter(MRJob):

    def mapper(self, key, line):
        temp = line.split('\t')
        title  = temp[0]
        text = temp[1]
        words = text.split(' ')
        count= Counter(words)
        for w in count:
            tf = count[w] * 1.0/ len(words)
            tf = str(tf)
            yield w, title + '  ' + tf

    def reducer(self, w, value):
        temp = '\t'.join(value)
        val = temp.split('\t')
        ls = []
        c = 0
        total_title = 196
        for m in val:
            c = c + 1
        for ele in val:
            temp2 = ele.split(' ')
            tf_idf = eval(temp2[-1]) * math.log(total_title * 1.0 / c + 1)
            ls.append([temp2[:-2], tf_idf])
        yield w, ls

if __name__=='__main__':
    MRFILE_TYPE_Counter.run()
```

参见：lab3.py

3. 实验结果

```
ics@ubuntu:~/tools/Python-3.6.2/mycode$ python lab3.py -r local ./MRresults.txt
```



实验成功结果，具体参加：output.txt

PartIII：文档查询函数

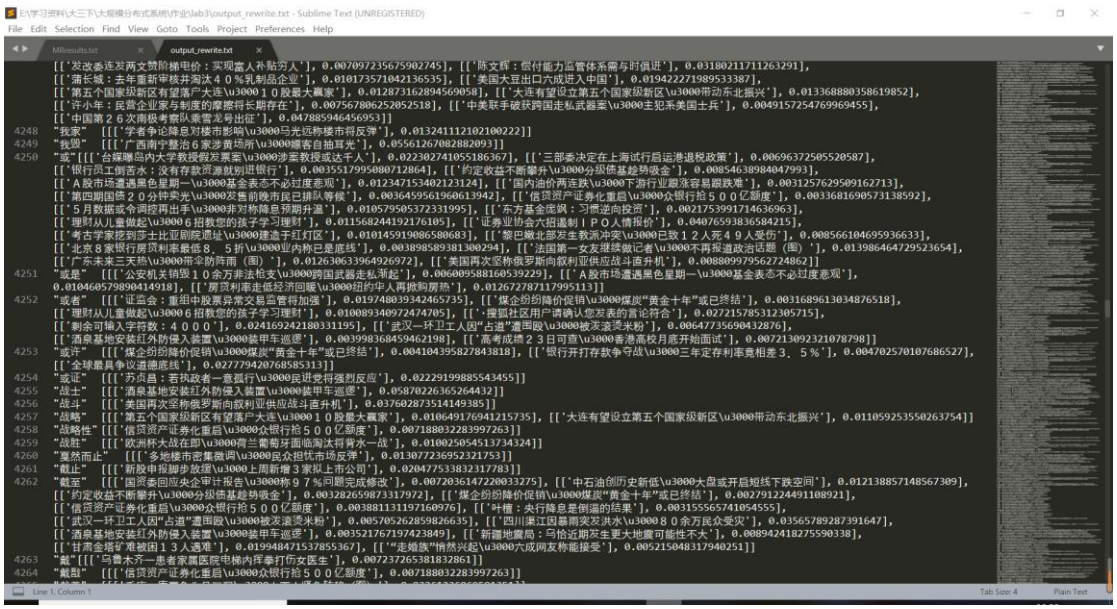
1. 由于输出是 unicode 编码，故而需要转化为中文。本地在线转换代码如下

```
# -*- coding: utf-8 -*-
"""
Created on Fri Apr  3 21:07:31 2020

@author: yuansiyu
"""

temp1 = []

address2 = "D:\output.txt"
with open(address2, 'r') as f1:
    for line in f1:
        line = line.strip('\n')
        temp1.append(line)
ls0 = []
for ele in temp1:
```

```
    a = ele.split('\t')
    a[0] = a[0].encode('utf-8').decode('unicode_escape')
    a[1] = eval(a[1])
    ls0.append(a)
f = open("D:/output_rewrite.txt", "w", encoding = 'utf-8')
for element in ls0:
    f.write(str(element[0])+'\t'+str(element[1])+'\n')
f.close()
```

参见：output2_rewrite.py

2. MR 输出结果



参见：output2_rewrite.py

3. 文档查询函数代码如下

```
# -*- coding: utf-8 -*-
"""
Created on Sat Apr  4 18:16:04 2020

@author: yuansiyu
"""
def sort_result(result):
    dic1 = {}
    for r in result:
        dic1[r[0][0]] = r[1]
    dic2 = sorted(dic1.items(), key=lambda x:x[1], reverse=True)

    print("查询结果为：\n")
    for r in dic2:
        print("网页名字：{}, 相关度：{}".format(r[0], str(r[1])))

if __name__ == '__main__':
```

```
temp1 = []

address2 = "D:/output_rewrite.txt"
with open(address2, 'r', encoding = 'utf-8') as f1:
    for line in f1:
        line = line.strip('\n')
        temp1.append(line)

dic = {}
for ele in temp1:
    a = ele.split('\t')
    a[0] = eval(a[0])
    a[1] = eval(a[1])
    dic[a[0]] = a[1]


print("请输入你想查询的关键词" + '\n')
key_word = input()
if key_word not in dic.keys():
    print("无相关页面显示，试试其他的关键词吧！")
else:
    result = dic[key_word]
    sort_result(result)

```

参见：file_search.py

4. 测验

(1) 查询“地震”关键词

```
In [30]: runfile('E:/学习资料/大三下/大规模分布式系统/作业/lab3/file_search.py', wdir='E:/学习资料/大三下/大规模分布式系统/作业/lab3')
请输入你想查询的关键词


地震
查询结果为：

网页名字：新疆地震局：乌恰近期发生更大地震可能性不大，相关度：0.236691114887194
网页名字：汶川地震三周年：“猪坚强”的幸福生活，相关度：0.08738942783450795
网页名字：探访非洲地狱般沙漠：火山毒气硫磺湖泊俱全，相关度：0.044624388681450876
```

(2) 查询“北京”关键词

```
In [33]: runfile('E:/学习资料/大三下/大规模分布式系统/作业/lab3/file_search.py', wdir='E:/学习资料/大三下/大规模分布式系统/作业/lab3')
请输入你想查询的关键词


北京
查询结果为：

网页名字：情人节当晚北京道路拥堵，相关度：0.15044723339456723
网页名字：北京８家银行房贷利率最低８．５折　业内称已是底线，相关度：0.029852521901914127
网页名字：北京宝马女挤翻本田撞飞路人　受审迟到庭上轻笑，相关度：0.029435328272850107
网页名字：国内油价两连跌　下游行业跟涨容易跟跌难，相关度：0.023934787130953875
网页名字：新股申报脚步放缓　上周新增３家拟上市公司，相关度：0.02099263721784659
网页名字：王耀辉涉农行副行长案　曾借道信托违规圈钱，相关度：0.011993136408778611
网页名字：中国渔政船正常巡航钓鱼岛遭日方“警告”，相关度：0.011283542504592542
网页名字：欧洲杯大战在即　荷兰葡萄牙面临淘汰将背水一战，相关度：0.01027723036471427
网页名字：父亲和三岁女儿翻唱《因为爱情》走红网络，相关度：0.009603014897525568
网页名字：监管部门开绿灯小产权房疯长　涉及群众一万余人，相关度：0.008231155626450486
网页名字：国资委回应央企审计报告　称９７％问题完成修改，相关度：0.006838510608843966
网页名字：多地楼市密集微调　民众担忧市场反弹，相关度：0.006703094557183689
网页名字：传汤灿被判１５年　经纪人回应称“在北京好好的”，相关度：0.006386910851656155
网页名字：药监局：６月底前完成城区餐饮单位量化分级管理，相关度：0.005320334383304931
```

(3) 查询“复旦”关键词

```
In [32]: runfile('E:/学习资料/大三下/大规模分布式系统/作业/lab3/file_search.py', wdir='E:/学习资料/大三下/大规模分布式系统/作业/lab3')
请输入你想查询的关键词


复旦
无相关页面显示，试试其他的关键词吧！
```