设计 MR 程序,实现具有 payload 的倒排文档索引

一. 实验介绍

mr job 是编写能够在 hadoop 上运行的 python 程序最简单的途径。如果使用 mr job,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, mr job 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. mr job 包

三. 实验过程

PartI:数据预处理

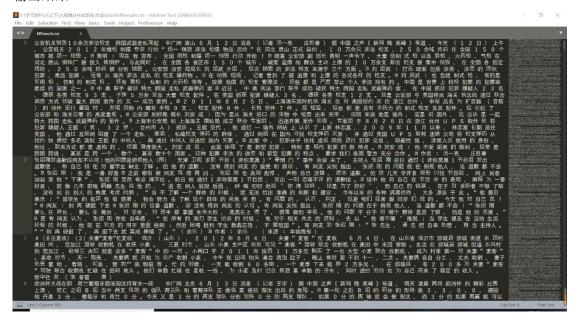
1. 将原文档用 jieba 库分词后预处理为〈网页名称 内容〉的格式文档。预处理代码 如下:

```
1. import jieba
    def write_file(r):
3.
      f = open("D:/MRresults.txt", "w", encoding = 'utf-8')
4.
      for key in r.keys():
5.
6.
        f.write(key + '\t' + r[key])
7.
        f.write('\n')
8.
      f.close()
9.
10. if_name_ == '_main_':
11.
      file = open('E:/学习资料/大三下/大规模分布式系统/作业
12.
    /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)
18.
19.
       for i in range(l):
         if "contenttitle" in ls[i]:
20.
            title = ls[i].replace('<contenttitle>',")
21.
            title = title.replace('</contenttitle>\n',")
22.
23.
            text = ls[i+1].replace('<content>','')
24.
            text = text.replace('</content>\n','')
            text = jieba.cut(text,cut_all=False)
25.
26.
            r[title] = ' '.join(text)
27.
28.
       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
3.
    import math
4.
    import os
    from collections import Counter
5.
6.
7.
    class MRFILE_TYPE_Counter(MRJob):
8.
9.
      def mapper(self, key, line):
10.
        temp = line.split('\t')
11.
        title = temp[0]
12.
        text = temp[1]
13.
        words = text.split(' ')
14.
        count= Counter(words)
15.
        for w in count:
16.
          tf = count[w] * 1.0/ len(words)
           tf = str(tf)
17.
18.
          yield w, title + ' ' + tf
19.
20.
      def reducer(self, w, value):
21.
        temp = '\t'.join(value)
22.
        val = temp.split('\t')
23.
        ls = []
24.
        c = 0
25.
        total_title = 196
26.
        for m in val:
27.
          c = c + 1
28.
        for ele in val:
29.
          temp2 = ele.split(' ')
30.
          tf_idf = eval(temp2[-1]) * math.log(total_title * 1.0 / c + 1)
31.
           ls.append([temp2[:-2], tf_idf])
32.
        yield w, ls
33.
34. if_name_=='_main_':
35.
      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

```
ics@ubuntu: ~/tools/Python-3.6.2/mycode
8e\u56fd\u540d\u6821"], 0.1367834522627081], [["\u80c6\u5c0f\u52ff\u5165\uff01\u
63ed\u79d8\u4e16\u754c\u4e0a\u6700\u7f8e\u60ac\u5d16"], 0.09321538969014183]]
"\uff53" [[["\u7559\u5b66\u5f55\u53d6\u7387\u6700\u9ad8\u7684\uff11\uff15
\u6240\u7f8e\u56fd\u540d\u6821"], 0.04559448408756937], [["\u80c6\u5c0f\u52ff\u5
165\uff01\u63ed\u79d8\u4e16\u754c\u4e0a\u6700\u7f8e\u60ac\u5d16"], 0.09321538969
\u6240\u7f8e\u56fd\u540d\u6821"], 0.06839172613135405], [["\u80c6\u5c0f\u52ff\u5
165\uff01\u63ed\u79d8\u4e16\u754c\u4e0a\u6700\u7f8e\u60ac\u5d16"], 0.09321538969
014183], [["\u82f1\u56fd\u6444\u5f71\u5e08\u8bb0\u5f55\u80d6\u4eba\u751f\u6d3b"]
014183], [["\u82f1\u56f
, 0.21511243774648114]]
 \u3000\u7ebd\u7ea6\u534e\u4eba\u518d\u6380\u8d2d\u623f\u70ed"], 0.03192268114041
081]]
"\uff5e'
"\uff5e" [[["\u7164\u4f01\u7eb7\u7eb7\u964d\u4ef7\u4fc3\u9500\u3000\u7164\u70ad\u201c\u9ec4\u91d1\u5341\u5e74\u201d\u6216\u5df2\u7ec8\u7ed3"], 0.00449620
3375865548], [["\u65b0\u7586\u5730\u9707\u5c40\uff1a\u4e4c\u6070\u8fd1\u671f\u53
d1\u751f\u66f4\u5927\u5730\u9707\u53ef\u80fd\u6027\u4e0d\u5927"], 0.014404764420
484608]]
Removing temp directory /tmp/lab3.ics.20200404.100641.809892...
 ics@ubuntu:~/tools/Python-3.6.2/mycode$ python lab3.py -r local ./MRresults.txt
 >> output.txt
No configs found; falling back on auto-configuration
No configs specified for local runner
Creating temp directory /tmp/lab3.ics.20200404.100713.511245
Running step 1 of 1...
job output is in /tmp/lab3.ics.20200404.100713.511245/output
Streaming final output from /tmp/lab3.ics.20200404.100713.511245/output...
Removing temp directory /tmp/lab3.ics.20200404.100713.511245...
```

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

```
1.
    # -*- coding: utf-8 -*-
2.
    Created on Fri Apr 3 21:07:31 2020
3.
4.
5.
    @author: yuansiyu
6.
7.
8.
    temp1 = []
9.
10. address2 = "D:\output.txt"
11. with open(address2, 'r') as f1:
12.
      for line in f1:
13.
        line = line.strip('\n')
14
        temp1.append(line)
15. ls0 = []
16. for ele in temp1:
```

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

参见: output2 rewrite.py

2. MR 输出结果

```
| Management | Ma
```

参见: output2_rewrite.py

. 文档查询函数代码如下

```
# -*- coding: utf-8 -*-
1.
2.
3.
    Created on Sat Apr 4 18:16:04 2020
4.
5.
    @author: yuansiyu
6.
7.
    def sort_result(result):
8.
      dic1 = {}
9.
      for r in result:
        dic1[r[0][0]] = r[1]
10.
      dic2 = sorted(dic1.items(), key=lambda x:x[1], reverse=True)
11.
12.
      print("查询结果为: \n")
13.
14.
      for r in dic2:
        print("网页名字: {},相关度: {}".format(r[0], str(r[1])))
15.
16.
17. if_name_ == '_main__':
```

```
18.
      temp1 = []
19.
      address2 = "D:/output_rewrite.txt"
20.
      with open(address2, 'r', encoding = 'utf-8') as f1:
21.
22.
        for line in f1:
23.
          line = line.strip(' \ n')
24.
          temp1.append(line)
25.
26.
      dic = \{\}
27.
      for ele in temp1:
28.
        a = ele.split('\t')
29.
        a[0] = eval(a[0])
30.
        a[1] = eval(a[1])
        dic[a[0]] = a[1]
31.
32.
33.
34.
      print("请输入你想查询的关键词" + '\n')
35.
      key_word = input()
36.
      if key_word not in dic.keys():
        print("无相关页面显示,试试其他的关键词吧!")
37.
38.
      else:
39.
        result = dic[key_word]
40.
        sort_result(result)
41.
```

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: **0.020**99263721784659
- 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 9 7 %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 1.5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: 0.006320334383304931

(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!