作业 5

钟诚 16307110259

April 17, 2020

设计 MR 程序，实现具有 payload 的倒排文档索引

1. 数据预处理

通过观察，我们发现待处理的数据为 HTML 格式的数据，如图 1。而当我们将数据输入 MRjob 进行处理的时候，要求数据按行输入。为了整理原始数据的编码格式和输入格式，我们需要对数据进行预处理。我们通过 process 函数，寻找数据中的新闻标题和正文，并将其对齐到一行，以 utf-8 的编码格式输出。代码及输入输出样例如下：

process.py:

```
def process(line):
    for line in lines:
        if line[0:14:] =='<contenttitle>':
            contenttitle =line[14:-16]
        if line[0:9:] =='<content>':
            content =line[9:-11:] # 对齐标题与正文
            f =open('data.txt','a+', encoding='utf-8')
            writelines =contenttitle +'\t' +content +'\n' # 在同一行进行输出
            f.write(writelines)
            f.close()
```

原始数据:

```
<doc>
<url>http://news.sohu.com/20120612/n345428229.shtml</url>
<docno>c172394d49da2142-69713306c0bb3300</docno>
<contenttitle>公安机关销毁１０余万非法枪支　跨国武器走私渐起</contenttitle>
<content>中广网唐山６月１２日消息（记者汤一亮　庄胜春）据中国之声《新闻晚高峰》报道，今天（１２日）上午，公安机关２０１２年缉枪制爆专项行动“统一销毁非法枪爆物品活动”在河
</doc>
<doc>
<url>http://news.sohu.com/20120607/n344998325.shtml</url>
<docno>dbb4554e49da2142-69713306c0bb3300</docno>
<contenttitle>张绍刚发道歉信网友不认可：他的问题是俯视他人（图）</contenttitle>
<content>天津卫视求职节目《非你莫属》“晕倒门”事件余波未了，主持人张绍刚前日通过《非你莫属》节目组发出道歉信，称自己错在对留学生缺乏了解。但他的道歉，没有得到网友的接受和
</doc>
<doc>
<url>http://news.sohu.com/20120604/n344745879.shtml</url>
<docno>3fca104f49da2142-69713306c0bb3300</docno>
<contenttitle>#（关注夏收）（３）夫妻“麦客”忙麦收</contenttitle>
<content>临沂（山东），２０１２年６月４日　夫妻“麦客”忙麦收　６月４日，在山东省临沂市郯城县郯城街道米顶村麦田间，范加江驾驶收割机在收获小麦。　三夏时节，山东小麦主产区处
</doc>
<doc>
<url>http://news.sohu.com/20120613/n345535702.shtml</url>
<docno>e4103f4f49da2142-69713306c0bb3300</docno>
<contenttitle>欧洲杯大战在即　荷兰葡萄牙面临淘汰将背水一战</contenttitle>
<content>中广网北京６月１３日消息（记者王宇）据中国之声《新闻晚高峰》报道，明天凌晨两场欧洲杯的精彩比赛上演，死亡之组Ｂ组当中两支传统的强队荷兰队和葡萄牙队正面临着提前淘
</doc>
<doc>
<url>http://news.sohu.com/20120601/n344598651.shtml</url>
<docno>ab18525249da2142-69713306c0bb3300</docno>
<contenttitle>扎克伯格携妻罗马当街吃３０元麦当劳午餐（组图）</contenttitle>
<content>环球网记者李亮报道，正在意大利度蜜月的“脸谱”创始人扎克伯格与他华裔妻子的一举一动都处于媒体的追踪之下。５月３１日，在罗马这个拥有１３家米其林星级餐厅的城市，身家
</doc>
<doc>
<url>http://news.sohu.com/20120606/n344833606.shtml</url>
<docno>e3736c5949da2142-69713306c0bb3300</docno>
<contenttitle>“十分钟”传导“三公里”</contenttitle>
<content>本报记者　张忠德　本报通讯员　张艳　苏婧　城区“十分钟”　从５月中旬开始，在胶南市珠海街道办事处的烟台东社区大舞台上，一台台京剧演出点燃了附近居民的热情，１００多
```

Figure 1: Rawdata

处理后数据，可见此时标题和正文已对齐到一行:

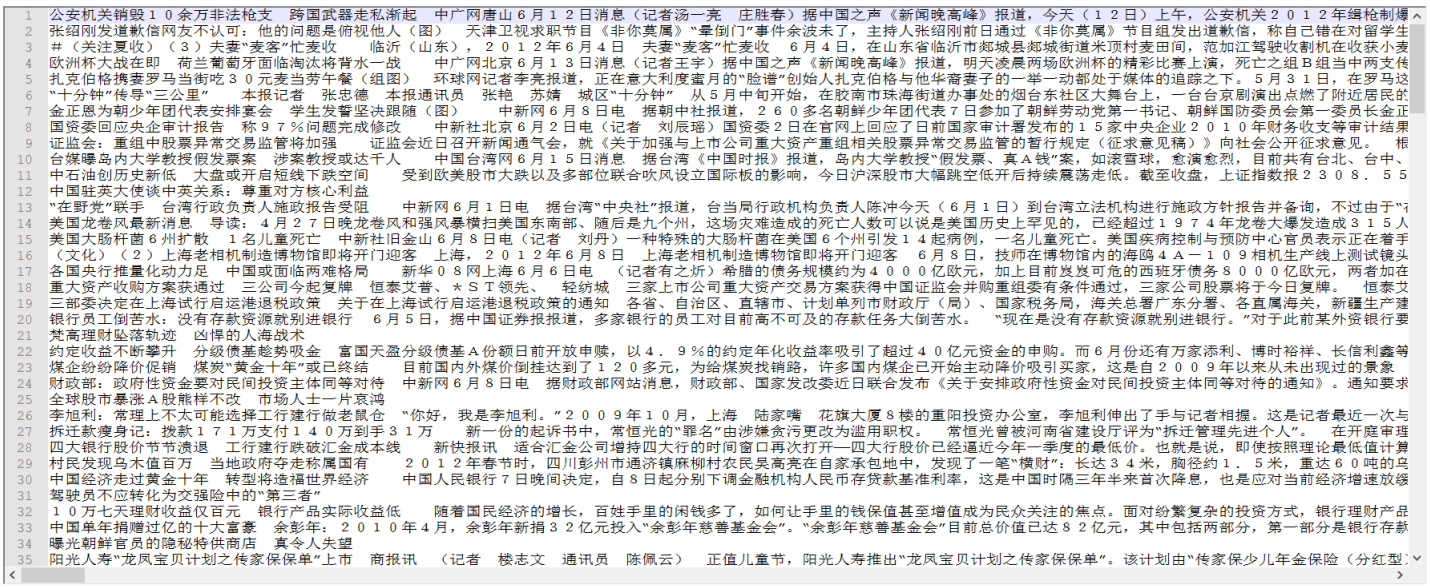

Figure 2: Processed data

2. 编写 MapReduce 程序并执行

随后我们对 MR 程序进行编写:

reducer.py:

```
from mrjob.job import MRJob
import jieba
class Reverse(MRJob):
    def mapper(self, _, line):
        contenttitle =line.split('\t')[0] # 提取题目
        SegmentDic ={}
        content =line.split('\t')[1] # 提取正文
        segs =jieba.cut(content) # 对正文进行分词
        for seg in segs:
            SegmentDic[seg] =SegmentDic.get(seg, 0) +1 # 对所有的分词进行计数
        for key,value in SegmentDic.items():
            yield key, [contenttitle, value] #输出为 (词语, [题目, 出现次数])

    def reducer(self, key, values):
        p = []
        for title, values in values:
            p.append((title,values)) # 对不同文章中出现的同一词语进行归并
        p.sort(key=lambda x: x[1], reverse=True) # 根据出现次数对文章标题进行排序
        yield key,p

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

然后，我们打开 HDFS 并执行 MapReduce 程序

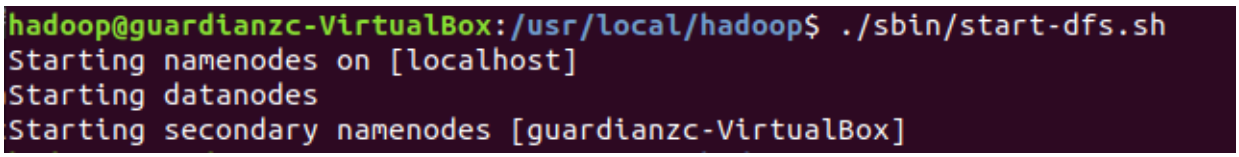

Figure 3: 打开 HDFS

```
hadoop@guardianzc-VirtualBox:~/文档/Project3$ python3 Project3.py -r local ./dat
a.txt >> result.txt
No configs found; falling back on auto-configuration
No configs specified for local runner
Creating temp directory /tmp/Project3.hadoop.20200410.123307.939124
Running step 1 of 1...
job output is in /tmp/Project3.hadoop.20200410.123307.939124/output
Streaming final output from /tmp/Project3.hadoop.20200410.123307.939124/output..
.
Removing temp directory /tmp/Project3.hadoop.20200410.123307.939124...
```

Figure 4: 本地执行 MapReduce

得到结果如下：

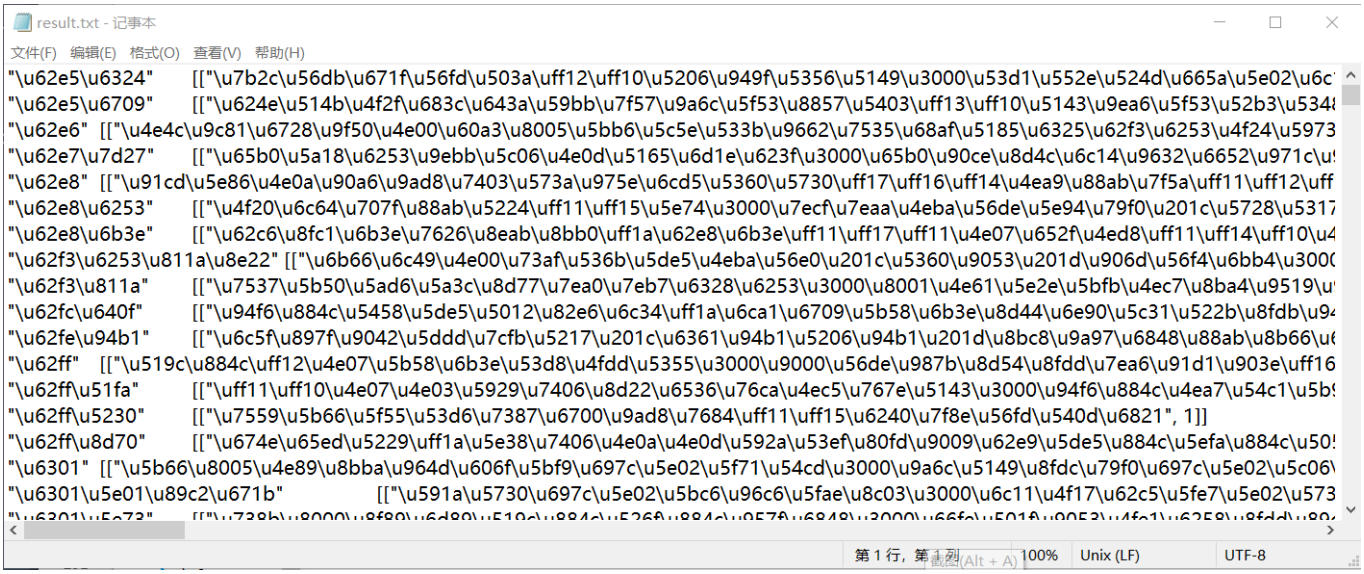

Figure 5: MapReduce 执行结果

值得注意的是：

1. 对于单个词来说，TF 即为词频，IDF 即为逆文档频率，而逆文档频率对于单个词语来说是相同的，故词频的多少即为该词语在文章中的 TF-IDF 大小。

3. 数据后处理

可见，MapReduce 输出结果仍需要调整编码，我以如下程序来解决这个问题：

coding.py

```python
def coding(file):
    file =open(file, encoding='utf-8')
    lines =file.readlines() # 读入文件
    for i in range(len(lines)):
        line =lines[i]
        if not file: # 如果读到文章底部即退出
            break
        line =line.encode('utf-8').decode('unicode_escape')
        with open('resultutf8.txt', 'a+') as f: # 在新文件中写入utf-8编码的数据
            try:
                f.write(line)
            except:
                pass
    file.close()
```

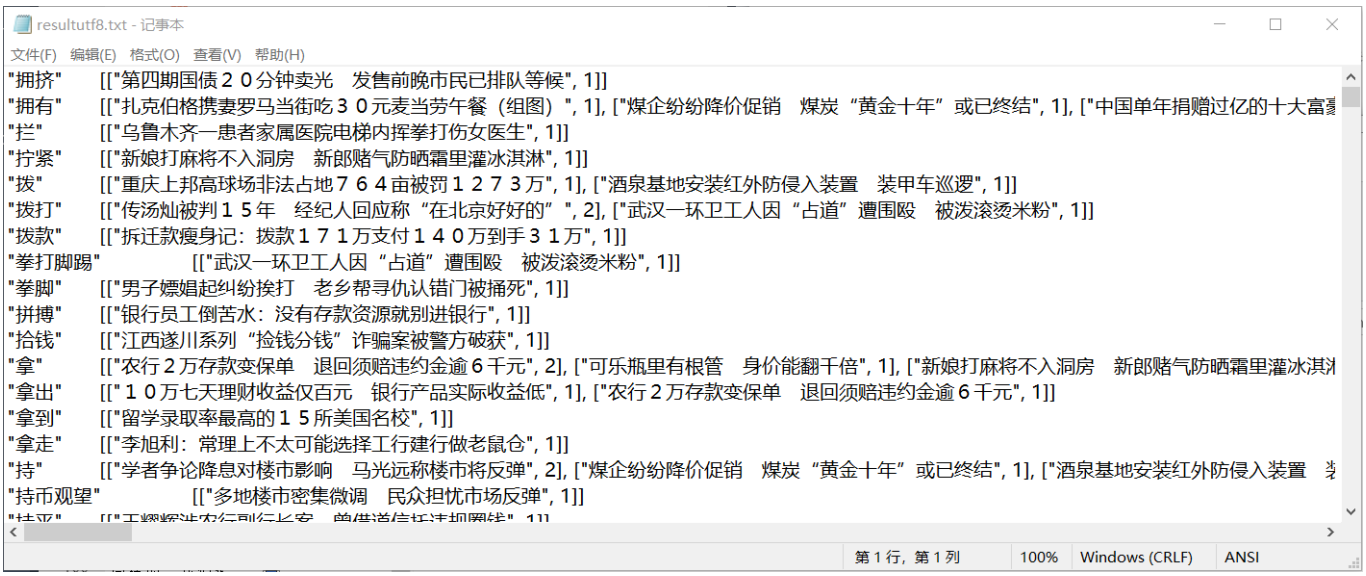

Figure 6: UTF-8 编码结果

这时，我们就可以对其中的词语进行倒排文档的查询

search.py

```
def search(word):
    file =open('resultutf8.txt','r')
    while 1:
        line =file.readline()
        if not file:
            break
        line_mark =line.split('\t')[0].rstrip('"').lstrip('"')
        if line_mark ==word: # 如果能查询到，则输出倒排文档值
            print(line_mark, '-->', line.split('\t')[1])
            return 0
    print('Not Found.') # 如果一直没有查询到，则输出“Not Found”
    return 0
```

以下为查询的示例结果：

```
(Normal) D:\Documents\课程介绍\大规模分布式系统\Project3\news_tensite_xml.smarty>D:/Application/Anaconda/envs/Normal/pyth
on.exe d:/Documents/课程介绍/大规模分布式系统/Project3/news_tensite_xml.smarty/transformer.py
控制 --> [["王耀辉涉农行副行长案 曾借道信托违规圈钱", 4], ["美国大肠杆菌６州扩散 １名儿童死亡", 3], ["证监会：重组中股
票异常交易监管将加强", 1], ["重大资产收购方案获通过 三公司今起复牌", 1], ["河南漯河省级文保旁建高楼 百年老房随时会坍塌"
, 1], ["缅甸若开邦骚乱局势总体得到控制", 1]]


(Normal) D:\Documents\课程介绍\大规模分布式系统\Project3\news_tensite_xml.smarty>D:/Application/Anaconda/envs/Normal/pyth
on.exe d:/Documents/课程介绍/大规模分布式系统/Project3/news_tensite_xml.smarty/transformer.py
接电话 --> [["可乐瓶里有根管 身价能翻千倍", 1]]
```

Figure 7: 查询结果

值得注意的是：

1. 这里对于单个词的查询，可以根据词频直接推出 TF-IDF 的值并排序，对于多个词的查询，则可以分别计算文档相似度并排序，与单个词的查询大致相同。

当然，如果需要实现多个词的查询，我们则要对所有文档和所有分词结果构建 TF-IDF 矩阵，然后对查询词的 TF-IDF 进行排序，具体实现如下：

query.py

```
import numpy as np
from math import log
word_dic ={}
```

```
counts_art ={}
# 构建词语对于每篇文章的计数矩阵
with open("./resultutf8.txt","r") as f:
    lines =f.readlines()
    idx =0
    for line in lines:
        word, titles =line.split('\t')
        word =word.rstrip("\"").lstrip("\"")
        titles =titles.rstrip(']]\n').lstrip('[[').split('], [')
        counts_word ={}
        for title in titles:
            t, count =title.split(',')
            count =int(count)
            t =t.lstrip('"').rstrip('"')
            # 词语计数
            counts_word[t] =counts_word.get(t, 0) +count
            # 文章总词数
            counts_art[t] =counts_art.get(t, 0) +count
        word_dic[word] =counts_word

art_name =list(counts_art.keys())

# 构建TF-IDF矩阵
art_totalnumber =len(art_name)
TF_IDF ={}
for key, value in word_dic.items():
    word_TFIDF ={}
    word_totalInArt =len(value.keys())
    for title, counts in value.items():
    # 前半为TF值，后半为IDF值
        word_TFIDF[title] =(counts /counts_art[title]) *log(art_totalnumber /( word_totalInArt +1))

    TF_IDF[key] =word_TFIDF

query =input("请输入查询的词语(空格分割):  ")
query =query.split()
# 计算并选出最高的五个结果
art_TFIDF ={}
for word in query:
    for art in art_name:
        art_TFIDF[art] =art_TFIDF.get(art, 0) +TF_IDF[word].get(art,0)
Max10 =sorted(art_TFIDF.items(), key=lambda item:item[1], reverse=True)[0:5]
print("最接近的5条文章为: ")
for high in Max10:
    print(high[0],'\t',high[1])
```

```
请输入查询的词语(空格分割):  改革
最接近的5条文章为:
日本首相野田组建新一届内阁  更换５名内阁大臣      0.03976166666174254
发改委连发两文赞阶梯电价：实现富人补贴穷人        0.01645310344623829
留学录取率最高的１５所美国名校    0.012965760867959523
中国经济走过黄金十年  转型将造福世界经济          0.006482880433979762
国资委回应央企审计报告  称９７％问题完成修改      0.006117179486421929
```

Figure 8: 查询“改革”

```
请输入查询的词语(空格分割):  开放
最接近的5条文章为:
脱衣舞“疯狂”农村夏夜让人汗颜      0.04827542705784238
大连有望设立第五个国家级新区  带动东北振兴        0.03186886382791551
第五个国家级新区有望落户大连  １０股最大赢家      0.030705218623362646
约定收益不断攀升  分级债基趁势吸金        0.03034776555440615
“在野党”联手  台湾行政负责人施政报告受阻          0.01088919407319753
```

Figure 9: 查询“开放”

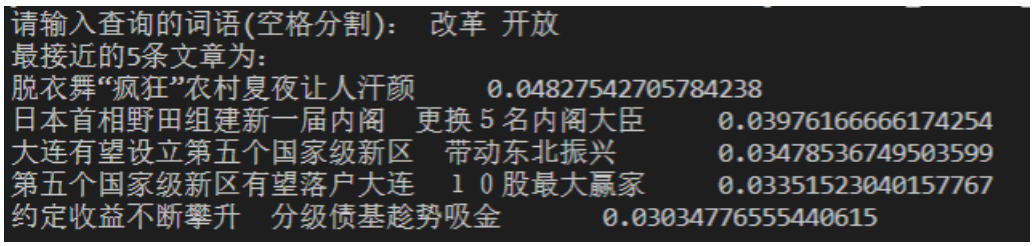

Figure 10: 查询“改革开放”

显然，当查询的词语更多时，查询的结果更加准确。如果需要进一步提升查询的准确率，则要求我们对文章的分词进行更加合理的处理。至此，我们完成了整个倒排文档查询项目。