FBDP_HW6

金融大数据作业6

姓名：盛祺晨　　学号：191220093

统计莎士比亚文集每个单词在各文档中出现次数。

设计思路

1.map

map端进行对文件的过滤大小写、去数字、去标点、过滤<3字数的单词、去常用词的过程，并将每个分出来的单词，打上标签1，写入context。基本和作业5方法一样。

其中，值得注意的是，这次为了将文件名加入，我先用getPath获取全称，并找到input（指定的输入文件夹）的位置。并获取那个txt文件的名字，然后将频率设定为Text("1")，写入context。

```
int splitIndex = split.getPath().toString().indexOf("/input");
keyInfo.set(curword+"#"+split.getPath().toString().substring(splitIndex+
7));//keyInfo例如 hadoop#file1.txt
valueInfo.set("1");
context.write(keyInfo,valueInfo);
```

2.combine

combine函数中，我将map到的“单词+url”的联合体拆分，然后让map中的valueInfo的数值加起来。让一个单词成为一个key，后面跟着一堆“URL+词频”

```
int sum = 0;
for (Text value : values) {
     sum += Integer.parseInt(value.toString());//将Text转换为int
}  // 统计次数
int splitIndex = key.toString().indexOf("#");
  // 重新设置value值由URL和词频组成
info.set(key.toString().substring(splitIndex + 1) + ":" + sum);
  // 重新设置key值为单词
key.set(key.toString().substring(0, splitIndex));
context.write(key, info);//key例如单词hadoop，info例如file1.txt:1
```

3.reduce

在reduce中，我用自建类DocCount 类来存储 文件名+次数 ， 用treeMap存储单词，字典序自动排序。

其中，treeMap部分是作业5的延续，可以将传入其中的key自动排序。只不过将key换成了String，并用compareToIgnoreCase的字符串比较方法变成字典序。

```
private TreeMap<String, DocCount> treeMap = new TreeMap<String,DocCount>
(new Comparator<String>() {
    @Override
    public int compare(String x, String y) {
    return x.compareToIgnoreCase(y); // 不区分大小写的字典序
    //                return x.compareTo(y); // 区分大小写的字典序
    }
});
```

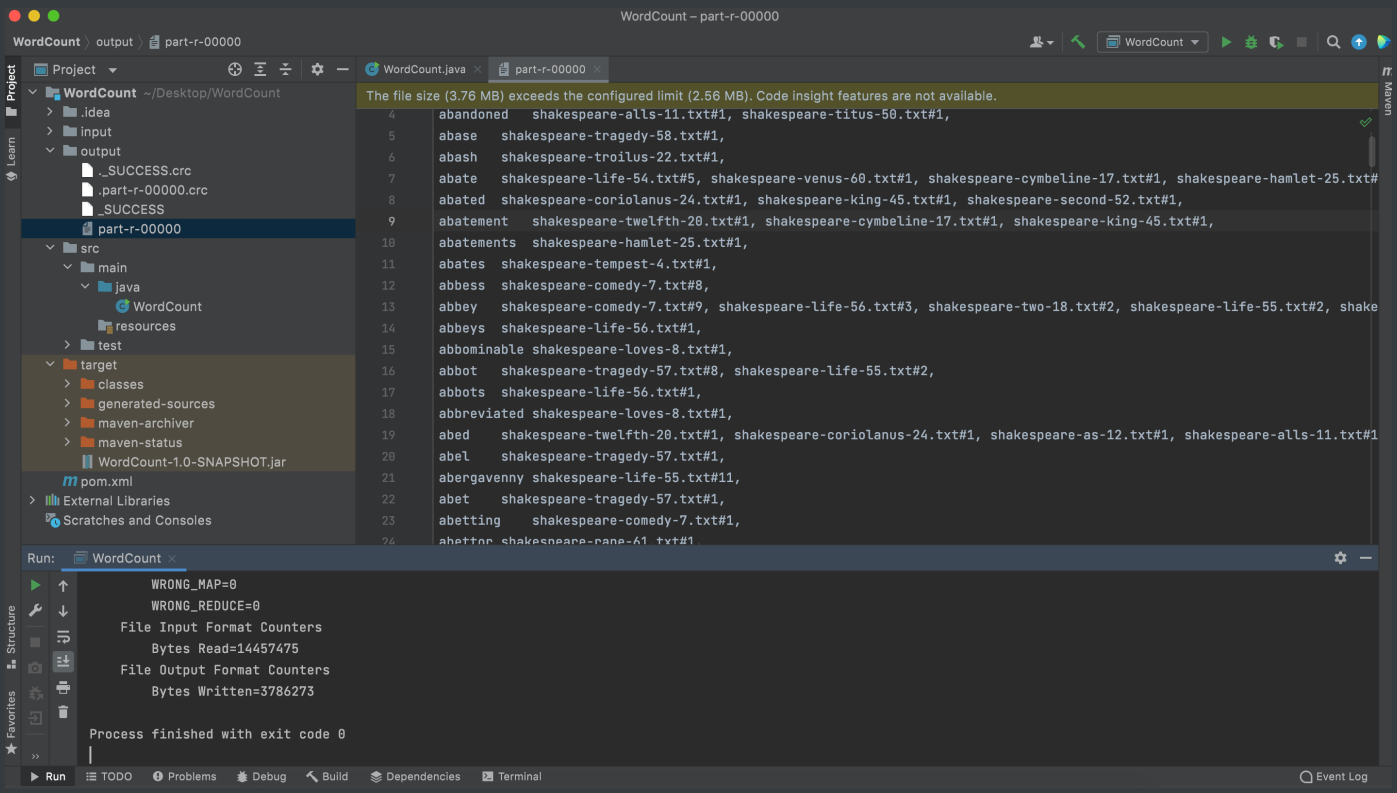
DocCount的形式如下：

```
public class DocCount{
    HashMap<String,Integer> docTimes = new HashMap<String,Integer>();
    DocCount(String setdoc,int setTimes){
        docTimes.put(setdoc,setTimes);
    }
    void addTimes(String Doc,int add){ // 给定Doc名字和sum，在Doc中增加sum次
count
        if (docTimes.containsKey(Doc)){
            docTimes.put(Doc, docTimes.get(Doc)+add);
        }
        else {
            docTimes.put(Doc,add);
        }
    }
    List<java.util.Map.Entry<String, Integer>> sort(){
        List<java.util.Map.Entry<String, Integer>> list = new
ArrayList<>(docTimes.entrySet());
        Collections.sort(list, new
Comparator<java.util.Map.Entry<String, Integer>>() {
            public int compare(java.util.Map.Entry<String, Integer>
entry1, java.util.Map.Entry<String, Integer> entry2) {
                return entry2.getValue() - entry1.getValue()
            }
        });
```

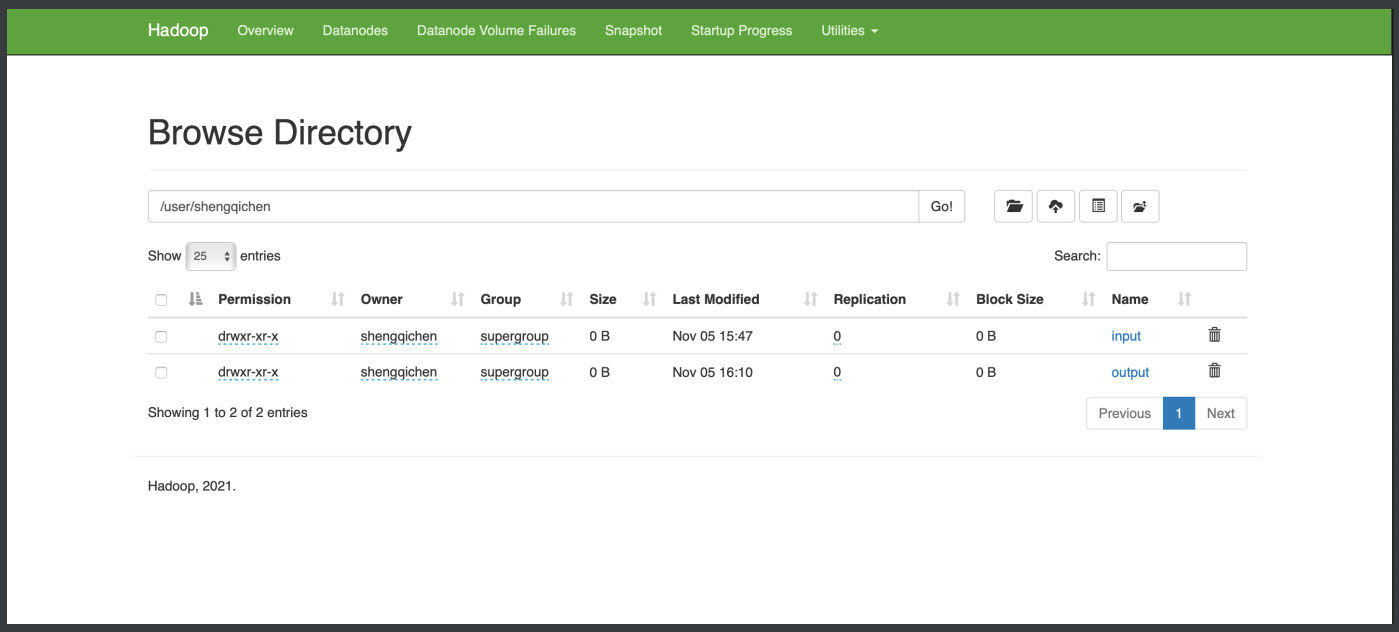
```
            return list;
        }
    }
```

结果展示

在IDEA环境中运行



在hadoop伪分布式上运行。打包，传参数做法如作业5.



用 hadoop dfs -cat ouput/part-r-00000 命令可以看出正确。

```
youthful        shakespeare-lovers-62.txt#658, shakespeare-two-18.txt#4, shakespeare-romeo-48.txt#3, shakespeare-second-52.txt#3, shakespeare-tragedy-57.txt#2, shakespeare-
sonnets-59.txt#2, shakespeare-troilus-22.txt#2, shakespeare-as-12.txt#2, shakespeare-comedy-7.txt#1, shakespeare-first-51.txt#1, shakespeare-merchant-5.txt#1, shakespeare-r
ape-61.txt#1, shakespeare-life-56.txt#1, shakespeare-alls-11.txt#1, shakespeare-merry-15.txt#1, shakespeare-julius-26.txt#1, shakespeare-third-53.txt#1, shakespeare-titus-5
0.txt#1,
youthfull       shakespeare-sonnets.txt#2,
youths  shakespeare-troilus-22.txt#1, shakespeare-pericles-21.txt#1, shakespeare-macbeth-46.txt#1, shakespeare-julius-26.txt#1, shakespeare-life-55.txt#1,
zanies  shakespeare-twelfth-20.txt#1,
zany    shakespeare-loves-8.txt#1,
zeal    shakespeare-life-56.txt#5, shakespeare-loves-8.txt#4, shakespeare-tragedy-58.txt#3, shakespeare-life-55.txt#3, shakespeare-second-52.txt#3, shakespeare-first-51.txt
#2, shakespeare-timon-49.txt#2, shakespeare-tragedy-57.txt#2, shakespeare-troilus-22.txt#2, shakespeare-two-18.txt#1, shakespeare-merchant-5.txt#1, shakespeare-life-54.txt#
1, shakespeare-much-3.txt#1, shakespeare-winters-19.txt#1, shakespeare-third-53.txt#1, shakespeare-titus-50.txt#1,
zealous shakespeare-life-56.txt#2, shakespeare-tragedy-58.txt#1, shakespeare-sonnets-59.txt#1, shakespeare-loves-8.txt#1, shakespeare-alls-11.txt#1,
zeals   shakespeare-timon-49.txt#1,
zed     shakespeare-king-45.txt#1,
zelous  shakespeare-sonnets.txt#1,
zenelophon      shakespeare-loves-8.txt#1,
zenith  shakespeare-tempest-4.txt#1,
zephyrs shakespeare-cymbeline-17.txt#1,
zir     shakespeare-king-45.txt#2,
zodiac  shakespeare-titus-50.txt#1,
zodiacs shakespeare-measure-13.txt#1,
zone    shakespeare-hamlet-25.txt#1,
zounds  shakespeare-first-51.txt#10, shakespeare-tragedy-58.txt#4, shakespeare-othello-47.txt#3, shakespeare-romeo-48.txt#2, shakespeare-life-56.txt#1, shakespeare-titus-50
.txt#1,
zwaggered       shakespeare-king-45.txt#1,
chengqichendeMacBook-Pro:WordCount shengqichen$
```

问题和解决

设计改进

1.路径相对写死，因为我传入的路径是固定的。其中args要穿入input和output，args[0]被我默认为input，要跳过的punctuation.txt和stop-word-list.txt是固定在input文件夹下的子目录skip。

```
    job.addCacheFile(new Path(args[0]+"/skip/punctuation.txt").toUri());
    job.addCacheFile(new Path(args[0]+"/skip/stop-word-list.txt").toUri());
```

改进：可以用GenericOptionsParser方法，把args都设定为可以用户自定义，这样会方便普适很多。

2.map中使用的是Text("1")，如果可以改为IntWritable就可以不在combine中重新将string变成int，花掉一些时间代价。