Assignment 2 独立完成部分

靳立晨 1600012459

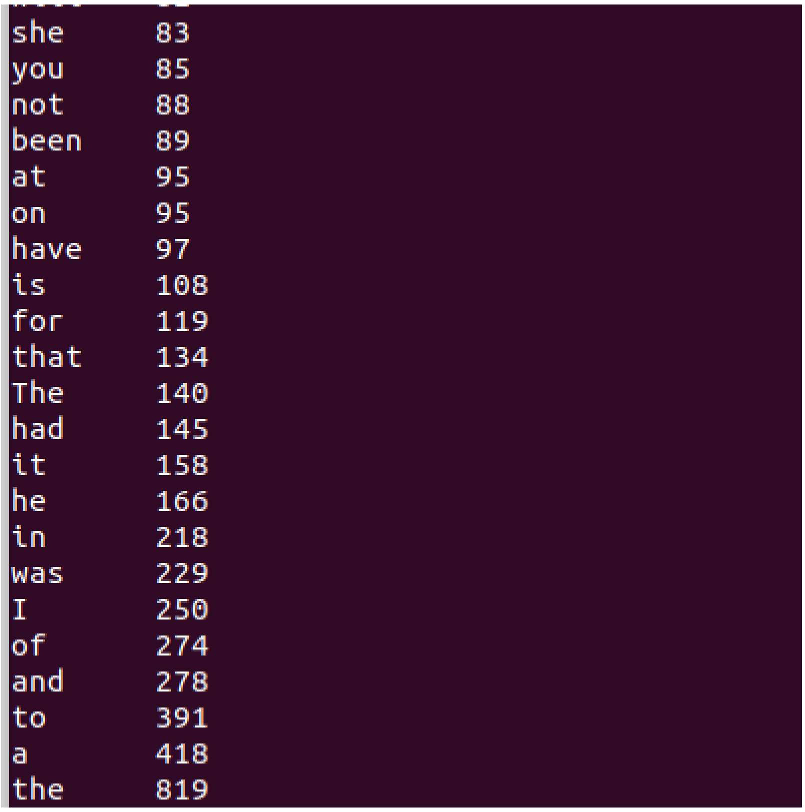
Task 1- 在新概念上进行Word Count

- 按照助教给的教程，在hadoop里面进行完整的Word Count练习。
- 发现果然存在大量标点符号，给结果带来了噪音。改造分词方式，使用正则表达式替换没用的符号，只保留字母、数字、分隔符和连字符。
- 结果整洁了许多，但是分隔符也作为引号出现在一些词的开头和结尾。结尾有像miles' 这样的正常情况不可区分，所以只选择去除开头的单引号。
- 代码如下：

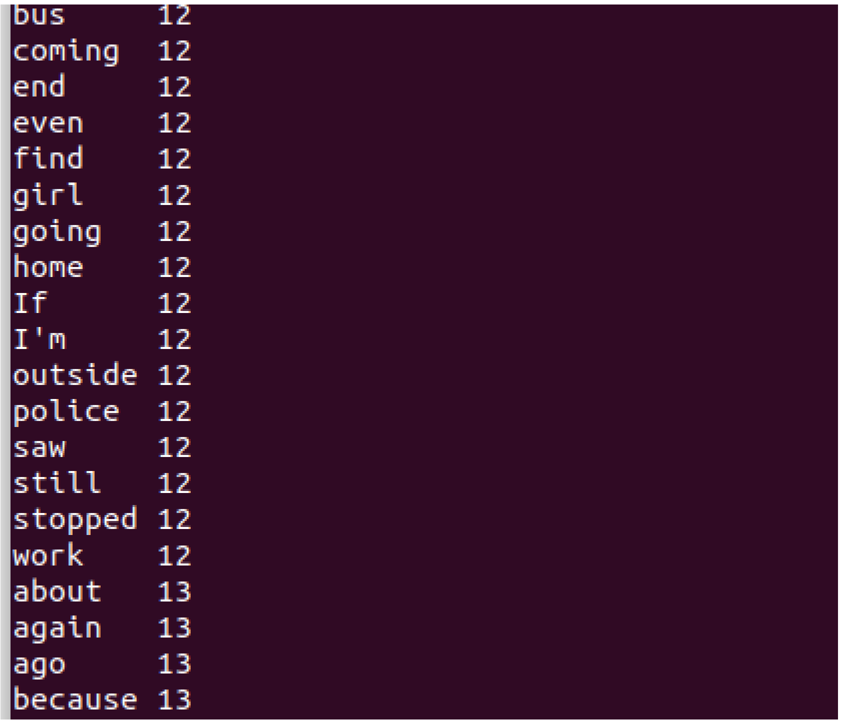
```
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCount {

  public static class TokenizerMapper
       extends Mapper<Object, Text, Text, IntWritable>{

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();
    private static String re = "[^A-Za-z0-9-']";
    public void map(Object key, Text value, Context context
                    ) throws IOException, InterruptedException {
      StringTokenizer itr = new StringTokenizer(value.toString().replaceAll(re, " "));
      while (itr.hasMoreTokens()) {
        String tmp = itr.nextToken();
        if(tmp.charAt(0) == '\'')
            tmp = tmp.substring(1);
        if(!tmp.isEmpty() )
            word.set(tmp);
            context.write(word, one);
      }
    }
  }

public static class IntSumReducer
```

```
       extends Reducer<Text,IntWritable,Text,IntWritable> {
    private IntWritable result = new IntWritable();

    public void reduce(Text key, Iterable<IntWritable> values,
                       Context context
                       ) throws IOException, InterruptedException {
      int sum = 0;
      for (IntWritable val : values) {
        sum += val.get();
      }
      result.set(sum);
      context.write(key, result);
    }
  }

  public static void main(String[] args) throws Exception {
    if(args.length!=2){
        System.err.println("Usage: wordcount <in> <out>");
        System.exit(2);
    }
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "word count");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```

- 结果如下图：



```
she       83
you       85
not       88
been      89
at        95
on        95
have      97
is        108
for       119
that      134
The       140
had       145
it        158
he        166
in        218
was       229
I         250
of        274
and       278
to        391
a         418
the       819
```



```
bus       12
coming    12
end       12
even      12
find      12
girl      12
going     12
home      12
If        12
I'm       12
outside 12
police    12
saw       12
still     12
stopped 12
work      12
about     13
again     13
ago       13
because 13
```

问题思考

- Mapper把文件以行/Block(太大时) 分开并行处理，这对于数据相互独立时没什么毛病；但要是行相关时就不会很方便了。小组作业4就会体现这一点。
- Combiner 在数据相关性强时或许不能执行和Reducer相同的运算，会打乱逻辑。