实验2报告

1. Map 和 Reduce 的设计思路
2. 必做部分：

Map部分(InvertedIndexMapper类)：从输入文档中提取单词，并记录每个单词出现在哪个文档中。

受课堂ppt上的参考代码启发，输出的value包括了文件名和行偏移量。同时考虑到输入文件中存在标点符号等既非字母也非数字的“非法”单词，在进行统计之前先将这部分符号从string中删除。

Reduce部分(InvertedIndexReducer类)：

Key为Text类型，包含单词和文档名(在mapper传入时使用’#’字符分割，并在reducer中使用split方法重新提取)；value为Text类型，表示单词的出现次数。

在这一部分会对单词的出现总次数word\_sum，以及在文档中出现的次数doc\_sum进行计算，并将结果暂存到叫posting的StringBuilder中，在cleanup时输出到context。

1. 选做1：top50高频词实现

Map部分(SortMapper)：

Reduce部分(SortReducer)：

1. 选做2：计算每个单词的TF-IDF

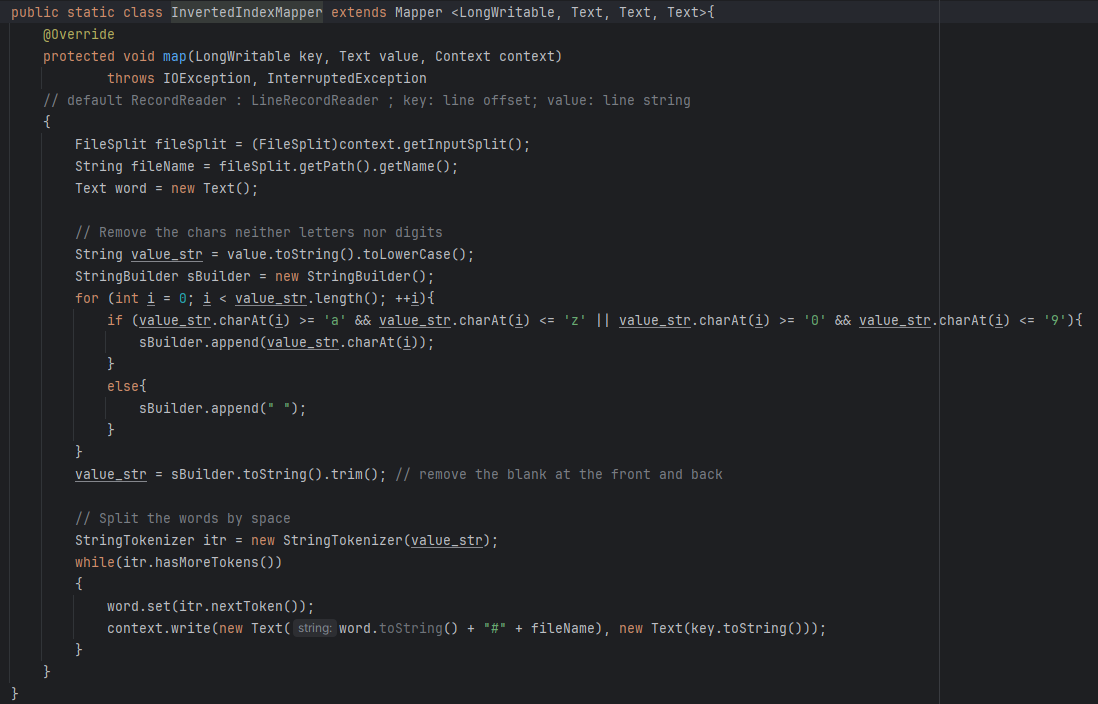
Map部分(TFIDFMapper)：

Reduce部分(TFIDFReducer)：

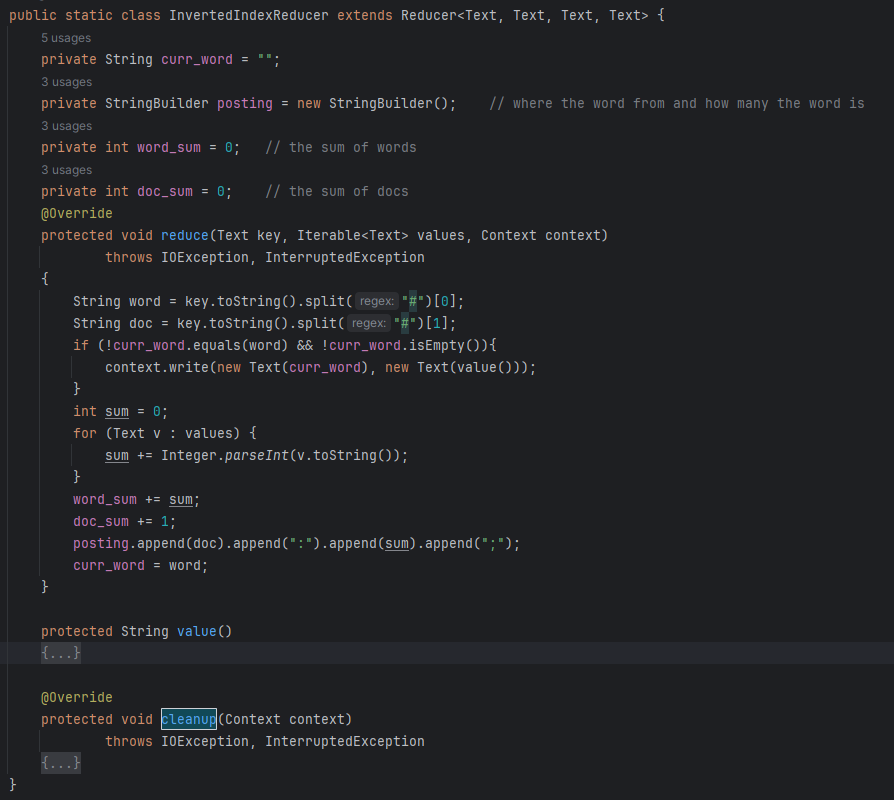
1. Map 和 Reduce 的代码(含注释)

介于篇幅有限，这里只展示必做部分的Mapper、Reducer与main函数部分：

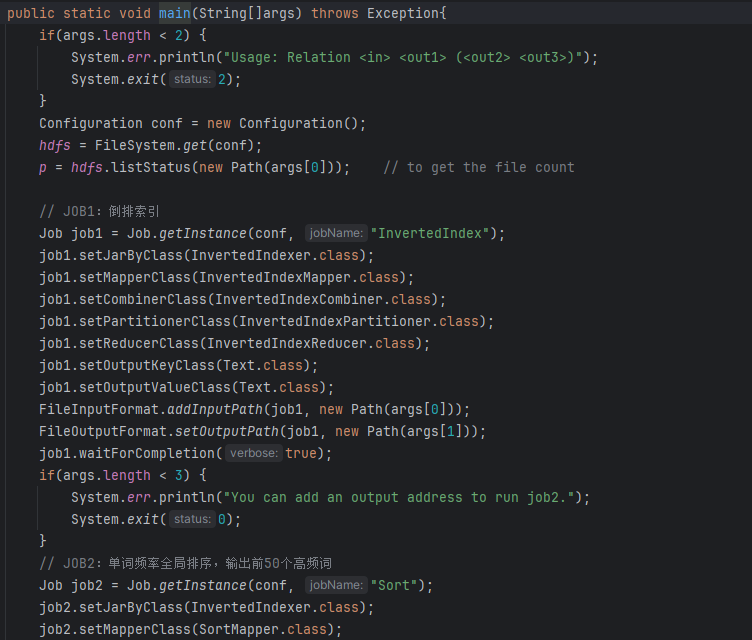
Mapper部分：



Reducer的实现：

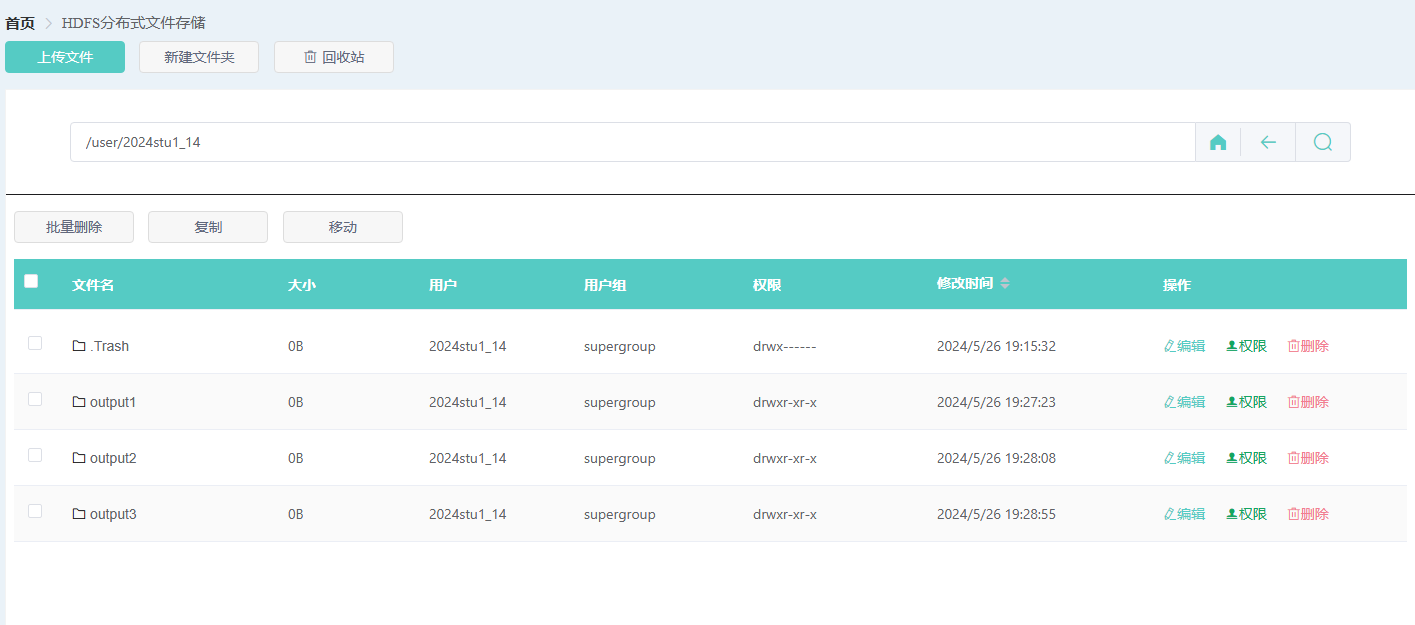


Main函数：

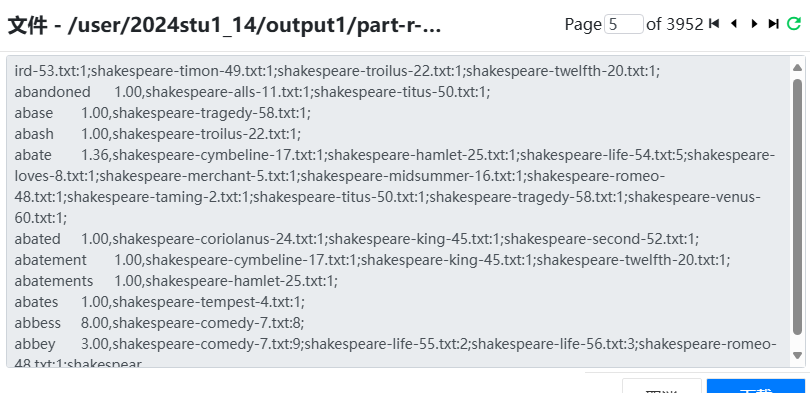


1. 输出结果文件的部分截图

针对必做和选做内容分别设置了输出目录，依次为用户根目录下的output1, output2 与output3。



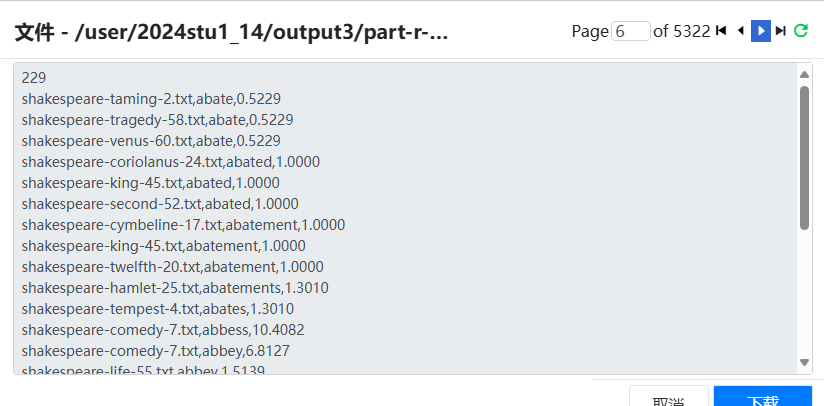
其中，output1中是倒排索引的输出，部分内容截图如下：



选做1输出排序后的前50个高频词，结果在output2中：



选做2为每个作品计算每个单词的TF-IDF，结果在output3中：



1. WebUI 执行报告内容

