基于文本内容的新闻检索与推荐

孙士杰 软件 71 2016011119

一．实验目的

本次实验是课程实验的第二部分，目标是根据之前所保存得到的分词结果和文档信息，建立倒排文档索引， 实现新闻查询及简单推荐的功能。

二．实验环境

开发环境:

操作系统：windows10

IDE：visual studio 2017 编程语言：C++

CPU：Intel® Core(TM) i7-6700HQ CPU @ 2.60GHz 2.59Hz

内存：8GB

三．抽象数据结构说明

本次实验主要实现了两种数据结构：AVLTree 和 DocLink,并对每种数据结构设计了相应的成员函数，都完成了实验要求中的功能。其中，AVLTree 中有类 AVLTreeNode，用于存放数据 key，左右孩子指针和文档链表 doc，AVLTree 有成员 head 指针以及所需的各种函数包括，插入、调整、查找等，参考了网上（见参考资料）的代码，实现了 LL,RR,LR,RL 四种旋转，便于调整。DocLink 中首先定义一个 info 类，便于存储数据和数据与结构的分离，此外，为了后续的方便，重载了 info 类的=（赋值）运算符；DocNode 是链表节点，有成员 info data 和 next 指针；DocLink 类有成员 head 指针和长度 len，此外，所有的函数也放在这个类中，包括添加，删除，遍历，复制，编辑等。

四．算法说明

这次实验主要在 AVL 树节点的增加上，参考课件和网上的代码，首先实现四种旋转和调整，再插入节点，根据字符串重载的字节比较进行大小判断，在插入的最后进行调整，并更新高度；doclink 中则是比较基础的链表操作，为了保证有序，在插入时便进行了排序，因为每次插入只针对一个节点，其他的是有序的，故若节点是无中生有，加在最后即可，若是已有节点的 num 增加，就依次和前面的比较，按需调整位置，同样对于 edit 函数（将相同文档编号的 num 相加，并保持有序）。

另外，在 AVL 中我为了做推荐，实现了一个 initconnection 函数，对于一个节点的词，对每一篇文章，计算其在分类中的重要性（在该文章中出现次数/在所有文章中出现次数），对于两篇文章，重要性相加即得到关联度，存放在一个 781*781 的 double 矩阵中。

此外还有 createAVL 函数（在 getfile.h）中，将分词结果按行读进来插入到 avl 树中。findArticle 函数，传入一个 CharString，将其中的空格去掉（removeBlank 函数，自己实现，新建一个字符串将其非空格字符复制过去，因为此前分词结果忽略了空格）后遍历数据库比对标题（**为保证搜索尽量有结果，采用 indexOf 比较，所以只要搜索的文字在这些文档的标题中存在即可比对上**），先比对上先输出。

实现两种功能的函数分别是 Query1 和 Query2，Query1 读入需要查询的词，按空格分开，在 avl 中寻找，找到后将对应文档链表复制到一个新的链表 p 中，最后输出到文件。Query2 中读入需要推荐的标题，去除空格后调用 findArticle 函数找到文档编号，将其关联矩阵中对应行取出，将文档编号和关联系数放进 info 数组中，进行排序，取最高五个输出到文件。

五．实验流程

本次实验采用之前的结果首先读进之前的所有分词结果建立 avl 树和倒排文档，再建立关联矩阵，再执行 Query1 和 Query2，得到结果。

Gui 因为输出有所不同，结构也略有不同。采用 qt 编写图形界面，将 query1，2 放在了 mainwindow 中输入文字后点击按钮出发相应的信号槽。因为 qt 的编码问题，我使用了一些临时文件来保存与转换输入。输入的要求与非图形界面的相同。操作方式见下或见/exe/readme.txt

六．操作说明

运行方式：将相应的 query1 和 query2 放到该目录中，双击 query.exe 运行即可

对于 gui，双击进入/gui 目录下，双击 gui.exe 运行即可

输入方式：query 双击后不需要输入

gui 在第一行的输入中输入需要查询的词，点击第一行的 search 键，即在下方可以看到推荐的文档标 题；在第二行的输入中输入要推荐的标题（一定要在数据库中），点击第二行的 suggest 键，即在下方可以看到推荐的文档标题和文档编号

七．实验结果

本次实验我采用了之前的分词结果，781 个文件分词时间约为 10min，因为采用流的方式读入，可能确实影响到了效率。

对于所要求的功能，经测试可以得到相对合理的结果.

八．功能亮点

九．实验体会

本次实验的难点在于 avl 树的平衡，因为是成熟的数据结构，所以各种资料非常齐全。经过自己的思考和参考各种资料能够做出来。此外难点便是 gui 了。由于编码的问题拖了很长时间没能解决，所以 gui 的效果不是很理想，最终解决编码的方式是先输出的文件再读进来，事实上很麻烦，但是我也没想到更好的方式。这次试验的难度还是比较大的，前后花费的时间很多，但是也有不小的进步，希望在今后的学习中能够更进一步。说实话觉得这里做 gui 的意义并不是很大，毕竟练习的重点应该还是在逻辑上和数据结构的练习上。

分词遇到的最大的问题是 325 号文档，因为其中有一部分长达可能数万个字符的乱码型的注释，我的分词算法会将其读入再抛弃，耗费了巨大的时间成本。

十．参考资料

https://blog.csdn.net/cjbct/article/details/53613436，
https://www.cnblogs.com/skywang12345/p/3576969.html，
https://www.jianshu.com/p/65c90aa1236d
https://blog.csdn.net/cjbct/article/details/53613436