数据结构课程实验报告

——基于文本内容的音乐检索与推荐（第二部分）

清华大学软件学院 李肇阳 2014013432

2015 年 12 月 17 日

目　录

§1　概述

§1.1　实验目的

实现一个基于文本的音乐检索与推荐系统。

对课堂上的经典数据结构进行练习；将数据结构知识应用到实际的软件开发中，体会数据结构的重要性和广泛用途；锻炼实际编程能力；等。

§1.2 开发环境

集成开发环境Visual Studio 2012 Premium (MSVC++ 11.0)，操作系统Microsoft Windows 7 Ultimate。

§2 实现说明

§2.1 主要数据结构和算法

链表、堆栈、字符串、字符串链表等数据结构直接沿用实验一。超文本解析、中文分词等算法直接沿用实验一。

§2.1.1 B-树（BTree）

模板类。保存key-value pairs，和std::map＜＞相仿。

实现了B-树的初始化、插入、查找、遍历操作。

§2.1.2 倒排文档（InvertedIndex）

此类维护了一张包含所有文档的文档链表，一棵用于保存关键词信息的B-树，还有一个分词器实例。

这棵B-树（如前所述，保存key-value pairs），key为关键词，value是一个结构，含有一张包含该关键词的文档的指针的链表，其中，每个文档指针都附加了该关键词在该文档中出现次数等的统计信息。

§2.1.3 搜索结果排序算法

按各关键词出现次数的加权平均值排序。对在标题出现的关键词赋予较高权重。对于词频超过预定阈值的词，权重置零（如“的、我、你”之类，剔除之以达到降噪的目的）。

§2.1.4 推荐算法

相当简单粗暴：直接将全文作分词后进行检索，截取检索结果的前10个。

§3 使用说明

§3.1 初始化流程

初始化倒排文档索引。

初始化分词器：程序读取iMusic.config，据其内容（“vocabulary.dic”）找到词典文件，加载词典。

读入并处理文档：遍历pages_300目录，对其中所有*.html文件依次执行：载入内存；解析；分词；加入索引。

§3.2 使用方法

§3.2.1 gui

图形界面。程序会监听本地端口2333，然后调用系统默认浏览器，打开`http://localhost:2333/index.html`页面以展示用户界面。

在文本框内输入关键词或歌曲名称后，按下回车或点击“查询”按钮，程序将相应显示检索结果或推荐结果。

结果列表中“歌曲名称”可以点击，点击后将展现该歌曲的推荐列表。

§3.2.2 query1

文本交互的索引功能。读取query1.txt并写入result1.txt。注意，输出的权值可直接用于搜索结果排序，但不代表真实原始的“出现次数”。

§3.2.3 query2

文本交互的推荐功能。读取query2.txt并写入result2.txt。

§3.3 注意事项

所有文件内容、文件名、目录名，均应统一采用GBK编码格式（ANSI简体中文 / Code Page 936）或其子集（GB2312、ASCII等）。程序不会读取字节序标记或HTML文档的charset元数据。

§4 实验结果

检索和推荐结果符合要求，不存在明显不当。

经测试，单个网页处理耗时0.01秒级（i5U 1.7GHz）。对300个网页，全过程内存占用峰值小于40MB。经性能分析，不存在明显性能瓶颈。

§5 得意之处

- 我的B-树对外提供遍历接口。
- 主要数据结构都是模板，可复用性强，安全可靠。实例化的时候好几层嵌套着都没问题，实际上我也是这么用的。
- 在执行检索时自动忽略词频极高的词以降噪。
- GUI上检索结果中关键词会高亮。歌词会截取显示含关键词的热点区域，
- GUI上“检索”和“推荐”两个功能有机统一。
- 界面构建、前后端交互方案（Web前端，ajax）是自己摸索出来的。自学并应用了一些C++11的新特性（lambda函数、自动类型推断、range-based for等）。
- 代码风格自认为不错（雾）假装写了单元测试（雾）

§6 感想与体会

实验一写得好就是好！指针安全，内存不泄露，功能封装合理，然后实验二就省了心了！哈哈哈哈

§7 致谢

感谢叶曦同学、唐人杰同学与我就本实验进行了不少有益的讨论。感谢陈凯助教。感谢git。感谢ThinkPad。

参考文献

[1] C++如何监听http请求. http://bbs.csdn.net/topics/390187516.

[2] unescape的C++实现. http://blog.csdn.net/zzstack/article/details/20556051.

[3] Google. Matetial Design Light. http://www.getmdl.io/.

[4] The jQuery Foundation. jQuery. http://jquery.com/.

[5] Thomas H.Cormen, Charles E.Leiserson, Ronald L.Rivest, Clifford Stein. Introduction to Algorithms (Third Edition). London: The MIT Press, 2009.

[6] 严蔚敏, 吴伟民. 数据结构（C语言版）. 北京: 清华大学出版社, 1997年4月.