数据结构课程实验报告

——基于文本内容的音乐检索与推荐

清华大学软件学院 李肇阳 2014013432

2015 年 10 月 26 日

目 录

§1 概述

§1.1 实验目的

解析HTML 文件提取歌曲的若干字段信息，并对歌词执行分词[1]。

熟悉经典数据结构、相关算法及其应用，提高编程能力。

§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。

[1]本文不是最终版本。本提交不是本作业的最终提交。

§2 实现说明

§2.1 主要数据结构

§2.1.1 链表

实现了一个双向不循环链表。模板类。为确保指针安全性以及避免内存泄露，全程采用深拷贝。

§2.1.2 堆栈

继承自链表。模板类。实现了堆栈的基本功能，包括压栈、退栈、取栈顶元素、检查栈是否为空等。

§2.1.3 字符串

动态管理的连续空间。实现了内存动态管理、KMP 模式匹配算法、增改字符、截取子串、拼接等。实现了与std::string 的双向转换。全程采用深拷贝。

§2.1.4 字符串链表

继承自链表模板以字符串进行的实例化。

§2.2 主要算法

§2.2.1 超文本解析

采用堆栈，查找“＜”、“＜／”、“＞”等具有特征的字符或字符序列以定位标签。将所有标签（丢失层次关系）保存在一张链表中。然后遍历该链表提取各项歌曲信息。

为减少特殊、不规则标签的影响，直接根据硬编码的特征字符序列切取热点区域，忽略其他部分。

接受HTML 字符串，返回解析好的歌曲信息结构。

封装在HTML 解析器类之中。

§2.2.2 中文分词

采用平凡的正向最大匹配算法（需预先提供词库，词库暂采用std::set 存储）。

对于英文单词（连续的ASCII 字符）进行了特别处理。具体来说，遇到ASCII 字符，检查其下一个字符，如果是空格或非ASCII 字符，则判定当前单词结束，发送到分词结果。

以配置文件的文件名进行初始化；接受字符串，返回字符串链表。

封装在分词器类之中。

§3 使用说明

§3.1 使用方法

命令行：iMusic arg1 arg2 arg3

其中arg1 为配置文件的文件名，arg2 为输入文件所在目录，arg3 为期待得到输出文件的目录。

配置文件只有一行，为词典文件的文件名。词典文件每行为一个词。

目录、文件名均应为绝对路径或者相对当前工作目录的相对路径。目录应是已存在的目录。

程序将遍历输入目录中的*.html 文件，处理后输出相应*.info、*.txt 文件。

所有相关文件均应采用ANSI（GBK、GB2312）编码格式。

§4 感想与讨论

感觉坑很多。有很多C++ 的知识是边写边复习的。课程实验（大作业）是个很好的形式，可以让我把所学的知识串联起来、应用起来，加深理解，加强工程实践能力。

作业刚刚布置时我曾对这种“重复发明轮子”的任务有些抵触，但完成之后，我感到对经典数据结构、经典算法、相关的工程上的问题的理解大大加深了。正所谓“绝知此事要躬行”，亲自动手实现一遍，比看书、看别人代码、直接调用标准库高到不知道哪里去了。总之是收获颇丰。

§5 致谢

感谢陈凯助教在实验全程及时回答我的疑问。感谢叶曦同学与我就本实验进行了不少有益的讨论。感谢git 这一由开源社区提供的方便的版本控制工具。感谢ThinkPad 品牌为我提供了很高的工作效率（小红帽大法好，Mac 和其他一切都是异端!）。

（完）

参考文献