数据结构第一次实验报告

姓名: 洪方舟

学号: 2016013259

班级: 软件62

E-mail: hongfz16@163.com

Tel: 15062727188

一、实验目标

- 1. 本学期数据结构课程实验希望实现基于挖掘机论坛网页抓取分析的销售线索检索
- 2. 第一次实验的基本目标包括实现数据结构: 栈、字符串、字符串链表
- 3. 使用自实现的数据结构实现网页的抓取与栈结构分析,提取关键内容
- 4. 使用自实现的数据结构实现文本内容的分词

二、实验环境

- 1. 基本功能模块开发(数据结构的实现与测试、栈结构分析网页、中文分词)
 - 操作系统: macOS High Sierra
 - 编译器: g++
 - 编程语言: C++
- 2. 编码转换模块
 - 操作系统: Windows7
 - IDE: Visual Studio 2012
 - 编程语言: C++
- 3. 各功能模块整合、测试、发布
 - 操作系统: Windows7
 - IDE: Visual Studio 2012
 - 编程语言: C++

三、抽象数据结构说明(仅列出重要成员)

- 1) 栈(类模板) class Stack;
- 1. 私有成员变量:

- T* ptrtop; 指向栈顶的指针
- int size; 栈当前的容量
- int num; 栈内元素的数量

2. 公有类方法:

- Stack();构造函数,初始栈空间为init_size
- void push(T ele); 向栈顶添加元素
- void pop(); 弹出栈顶元素
- T top(); 返回栈顶元素值

2) 字符串 class CharString;

1. 私有成员变量:

- int size; 申请的空间大小
- int num; 字符串的长度,即使用的空间大小
- char* base; 指向字符串的指针

2. 公有类方法:

- CharString(***); 构造函数,可以使用的参数如下: 空(默认申请init_size的空间)、char、char, int、string&、char*
- CharString(const CharString& cs); 拷贝构造函数, 防止编译器生成的默认拷贝函数的直接复制指针的行为
- int indexOf(CharString& targetString,int pos);返回从pos开始的第一个匹配targetString的位置的index(pos从0开始数、以下表示index的都是从0开始数)
- CharString subString(int pos,int len);返回从pos位置开始长度为len的字符串(包含第pos个)
- CharString* concat(CharString& targetString);将targetString连接到本身之后,返回一个连接好的字符串的指针,这个操作并不会改变本身或者targetString
- void assign(CharString& targetString); 赋值, 将targetString赋值给this
- void append(***); 将某个字符(串)添加到末尾,可传入的参数如下: char、char*、string&、CharString&
- char getIndex(int index) const;返回第index个字符
- bool is_equal(CharString target) const; 如果两字符串相同返回true, 否则返回false
- friend std::ostream& operator << (std::ostream& out,CharString& obj); 重载标准输出流操作符

3) 双向循环链表(类模板)class Link;

1. 私有成员变量:

- Telem; 当前节点存储的信息
- Link* prev; 前一个节点的指针
- Link* next; 后一个节点的指针

2. 公有类方法:

- Link();构造函数,初始化操作,将prev和next指针指向自己
- void add(T& elem); 向链表最后添加一个节点

- T remove(int index); 将第index个节点删除
- T remove(Link* rm); 将rm指向的节点删除
- Link* search(T& target); 查找链表中是否有值等于target的节点,如果有就返回该节点指针,否则返回nullptr
- void insert(Link* pos,T& elem);在pos节点之前插入一个节点,元素为elem
- Link* getNode(int index);返回第index个节点的指针
- void printLink(std::ostream& os); 打印链表
- T getElem();返回当前节点存储的数据

4) Unicode到GBK转码类class UnicodeToGBK;

1. 私有成员变量

- std::map<int,int> uni2gbk; unicode与GBK的编码对应表

2. 公有类方法

- UnicodeToGBK(); 默认构造函数,读取UnicodeToGBK.csv文件,初始化Unicode与GBK的对应std::map
- ~UnicodeToGBK(); 析构函数, 释放空间
- int* convert_int(char* unicode, int length 传入的字符串格式为"&#xxxxx;...", 返回值为每个中文字对应的两个窄字符的取值
- int* convert_single(int uni); 传入的为unicode编码,返回值为该编码的中文字对应的两个窄字符的取值

5) 中文分词类

1. 私有成员变量

- std::set<CharString> dic; 存储普通词典的词条
- std::set<CharString> super_dic; 存储专业词汇词典的词条

2. 公有类方法

- wordSegmentation(string filename, string super_filename); 初始化词典,第一个参数为普通词典,第二个参数为专业词汇词典
- ~wordSegmentation(); 析构函数, 释放空间
- Link<CharString>* divideWords(CharString& incstring, int max); 参数为待分词的字符串以及 逆向最大匹配法中的最大匹配字符数,返回一个字符串列表

四、算法说明

1. 网页解析

从头开始遍历存储在CharString类型中的html文件,每当遇到<就进入函数label get_label_info(CharString&html,int pos);,返回当前标签的信息(标签类型,class值),将返回的label压栈。如果遇到目标标签(如<div class="z">) 就进入相应函数提取信息(如int find_basic_info(Stack<label>& m_stack,CharString&html,int pos,CharString* info); 传入参数为栈的引用,字符串引用,当前遍历到的位置,存储提取信息的字符串数组,返回值为目标标签闭合位置)。如果遇到</或者/>的结构就调用CharString get_close_info(CharString&

html, int pos); ,传入待分析字符串引用和当前遍历位置,返回该闭合标签的信息,与栈顶的标签进行匹配,如果匹配成功就弹出栈顶。遍历结束时提取信息完成。

2. 中文分词

采用逆向最大匹配法。从句子末尾开始,每次从目标句中取出长度为len的词组,在词典中进行匹配,如果成功则从成功匹配的词组前一个字重复匹配过程,如果失败则取出长度为len-1的词组,在词典中匹配。如果len==1直接将这一个字单独作为分词的一个部分。

在匹配时优先匹配专业词汇, 再匹配普通词典。

在处理中文的时候有一点要特别注意,每一个中文字符占2个char,所以每次要取出长度正确的CharString保证里面存储的是完整的中文字。

3. Unicode转GBK

包装成一个类class UnicodeToGBK;从网络上找到一张Unicode与GBK的映射表,在初始化的时候使用std::map加载映射表。将待转换的字符的Unicode编码传给类方法,取出对应的GBK编码。将对应的GBK编码高八位和低八位取出来分别存入int型中,返回两个int型。在使用的时候需要将这两个int型转为char型,高八位对应的char在先低八位对应的char在后依次放入CharString中。

五、流程概述

- 1. 初始化词典和Unicode到GBK转换表
- 2. 从待解析url列表中获取一个url,获取页面源代码后存入暂时文件中。
- 3. 从暂时文件中读取html源代码存入CharString中, 开始栈结构解析页面
- 4. 将提取内容中的Unicode编码部分转换为GBK编码
- 5. 将标题部分和发帖内容做中文分词
- 6. 将所有信息输出到输出文件中

六、输入输出及相关操作说明

- 1. 请将输入文件url.csv放置在exe/input文件夹下,输入文件格式请和示例文件一致,否则将导致输入错误。
- 2. 放置好输入文件后直接运行Experiment1.exe可执行文件。
- 3. 请在在exe/output文件夹下读取输出结果。

七、实验结果

- 1. 给定的100个网页的解析结果表明,发帖大类,发帖小类,发帖标题,发帖内容,发帖人,发帖日期,发帖类型的解析结果出色,基本实现了关键信息的无遗漏提取。
- 2. 通过逆向最大匹配法,使用助教所给词典得出的结果并不十分理想,一方面是因为发帖内容的语言比较零碎,口语化;另一方面是因为词典的词组具有局限性。在使用了自己的词典(功能亮点部分将会介绍)后结果稍微理想一些。

八、功能亮点说明

- 1. 通过python爬虫爬取20000个帖子的文本内容,进行分词词频统计,将高频词提取后扩充词库。具体代码与词频结果请参见src/scraping文件夹下README.md文件。
- 2. 通过GBK与Unicode对照表自实现了转码器。

九、实验体会

- 1. 网页分析部分和数据结构实现其实并没有什么难度,也很快就写完了。
- 2. 困难的部分在于网页中文本编码不一致造成的处理困难,以及该网站的不同帖子有时会有不同的html结构,导致了处理上的一些麻烦。
- 3. 中文分词在不介入机器学习,词库不理想的情况下,很难做到比较理想的结果,鉴于本人对于机器学习并不熟悉,所以选择了通过抓取大量的网页数据进行词频统计扩充词库。
- 4. 个人有一点建议,建议之后可以找些外国论坛或者是文本编码比较一致的中文论坛,减少同学们耗费在文本编码处理上的时间,将时间更多的放在分词算法,网页结构解析算法的优化,以及更多有趣的数据分析上。