实验报告

基于文本内容的销售线索检索（第一部分）

一、实验目标

从指定的100个网页中提取关键信息，针对提取信息中的发帖标题和发帖内容使用分词算法进行分词，并将提取的信息和分词结果保存到文件中。

二、实验环境

* 操作系统：Windows 10 64位（DirectX 12）
* CPU：英特尔Core i7-6700HQ @ 2.60GHz 四核
* 内存：8GB
* IDE：Visual Studio 2015

三、抽象数据结构说明

1、自定义字符串

* 在CP\_MyString.h文件与CP\_MyString.cpp文件中定义了myString类，实现了自定义字符串的相关操作。
* myString类中主要成员函数介绍

|  |  |
| --- | --- |
| 函数名 | 函数功能 |
| IndexOf | 从指定位置开始搜索，返回子串在主串中的位置；若主串中不存在子串，则返回-1。 |
| SplitString | 将自定义字符串以给定字符串为标识进行分割，将分割结果存储在自定义字符串向量中。 |
| SubString | 进行字符串拷贝，从目标字符串的指定位置开始拷贝指定长度。 |
| ConcatString | 进行字符串的连接。 |

* myString类中重载的运算符介绍

|  |  |
| --- | --- |
| 重载的运算符 | 运算符功能 |
| = | 可以将string或者myString赋值给myString对象。 |
| + | 可以在myString对象后连接字符。 |
| []（下标运算符） | 可以返回myString对象对应位置的字符。 |

2、自定义字符串链表

* 在CP\_MyStringList.h文件与CP\_MyStringList.cpp文件中定义了myStringList类，实现了自定义字符串链表的相关操作。
* 链表的结点为结构体myStringNode，其中包括myString对象作为存储数据，以及指向下一个结点的指针\*next。
* myStringList类中主要成员函数介绍

|  |  |
| --- | --- |
| 函数名 | 函数功能 |
| addNode | 将给定的myString类对象连接到链表末尾。 |
| removeNode | 移除链表中数据为给定myString对象的结点，若该结点不存在则直接返回。 |
| searchNode | 寻找某一结点的位置，若不存在该结点则返回0。 |

3、自定义栈结构

* 在CP\_MyStack.h文件与CP\_MyStack.cpp文件中定义了myStack类，实现了自定义栈结构的相关操作。
* 栈结构压入的为结构体stackData，其中包括存储标签的myString对象以及整数的标签编号。
* myStack类中主要成员函数介绍

|  |  |
| --- | --- |
| 函数名 | 函数功能 |
| Init | 初始化栈。 |
| Push | 压入对应标签的stackData。 |
| Pop | 弹出栈顶的标签。 |
| Top | 获取栈顶元素，若栈为空则因程序退出。 |
| Empty | 判断栈是否为空，返回true/false。 |

四、算法说明

1、网页信息获取

利用CInternetSession和ChttpFile类对象与指定网页连接，并将网页源码读取到myString的向量中进行存储。

2、网页信息解析

1. 对于提取到的网页内容进行遍历，若遇到标签符号则将其压栈。
2. 返回栈顶的标签，判断是否为特殊标签字符串。若是特殊标签字符串，则读取其中的内容并存储到字符串链表中，否则继续遍历，直至遍历完所有标签，栈为空。
3. 对提取的内容进行处理，将unicode编码的汉字或符号进行转化，并修改其中的英文逗号避免错误输出，从而将提取的信息转化为正规语句。

3、中文分词算法

1. 利用系统自带哈希表，导入词库与专业用语库创建哈希词典。
2. 对内容与标题的字符串进行预处理，去除其中的符号，只保留汉字。
3. 利用逆向最大匹配法对字符串进行分割与匹配筛选，将分词存入字符串中。

五、实验流程

1. 进行联网操作，确定程序可以正常读取网页。
2. 在input文件夹中放入url.csv文件，运行程序。
3. 程序运行过程中会有进程提示，操作完成会提示“已输出结果。”，此时可以关闭程序，检查output文件夹中的result.csv的输出结果。

六、操作说明

1. 务必确定网络正常连接，否则程序会报错。
2. 请提前确认所在网页域名正确且存在目标信息，否则无法获取正确结果。

七、实验结果

在result.csv文件中会输出一百个网址的发帖大类、发帖小类、发帖标题、发帖内容、发帖人、发帖日期、发帖类型与分词结果的信息。

八、功能亮点

* 压栈时会给不同的标签赋予不同的编号，如<div>为1、<a>为2等，在出栈时判断后标签与前标签是否匹配，从而避免后标签多余导致的出栈错误。
* 在程序运行时会提示运行进程，从而确定程序是否正常运行。

九、存在问题

* 关键信息提取算法较为暴力，效率较低且错误率较高。
* 分词算法未实现优化。
* 在无网络连接、提取信息错误时没有很好地应对措施，有可能导致程序崩溃或结果出错，鲁棒性有待提高。

十、实验体会

这次实验是一个思考问题解决问题的过程，在实验中我知道了现实中实际问题的复杂性，同时也发掘出这一学期所学习的各类数据结构的许多功能。我一开始写好的程序在运行过程中会占用多达200MB内存，在我不断优化与解决了内存泄漏问题之后，程序运行仅需要30多MB内存，这一点也让我切实感受到好的算法的重要意义。