实验报告

1 实验目标

“实验希望实现一个基于文本内容的知乎日报文章检索”……蛤！？，不能复制助教的啊！那我就用自己的话概括一下：本次实验的目标就是高举“造轮子大法好（雾）”的旗帜，坚定不移地贯彻挖大坑的装逼路线，努力用最繁琐的方式实现本来可以百来行搞定的事。

注：本项目也发布在 https://github.com/sunziping2016/zhihu-search-engine。

2 实验环境

我可以回答一句“无可奉告”，但是助教可能不开心，我怎么办？……咳咳，这次我才用了混合开发环境，Linux 上为 **Arch Linux x64 + GCC 6.2.1 + Node.js 7.2.1 + Electron 1.4.12 + CLion 2016.2.3**，Windows 上为 **Windows 10 x64 + Visual Studio 2017 RC + Node.js 7.2.1 + Electron 1.4.12**。

代码中广泛使用了 C++ 11 的语法，包括但不仅限于 move 语义和多线程。

除了 mydir.h 和 mydir.cpp 采用了平台相关的代码，别的组件均为平台无关。目前 mydir.h 和 mydir.cpp 只支持 POSIX（Mac OS、Linux、Unix 等等）和 Windows NT 系列的平台。

3 抽象数据结构说明

助教：不使用标准库吼不吼啊？
我：吼啊！
助教：那么，你的代码泛型资不资瓷啊？
我：当然资瓷啊！
……
助教：那些个 AVL 吧！
我：雾草，我辣鸡不会（逃） ε = ε = ε = ┏(゜ロ ゜;)┛

3.1 AVL

myavl.h 中有一个非常辣鸡的 AVL 泛型类，class myavl。支持**插入删除搜索**，支持**自动扩容**。（没有写迭代器，实际上也没怎么用这个类）

3.2 文档链表

啊，忘记写了，要不在这里写吧：

```
typedef mylist<zhihu_content> a_useless_wen_dang_lian_biao_class;
```

3.3 前缀树

唔，新写了个每层都是哈希表的前缀树，在“trie.h”和“trie.cpp”中。只是为了性能。

4 算法说明

4.1 解析和建树算法

相比上次实验采用了线程池的多线程算法。其中线程池“threadpool.h”这个类来自 GitHub 开源项目（虽然我还能写出更强的，逃）。还是为了性能。

分词做了改进：包括对于标点、空白符处理的优化，英文分词等等。

4.2 检索算法

图形界面的检索算法与命令行界面的检索算法略有不同。图形界面对文章的标题同样进行检索，且对标题检索得到的结果给予了更高的权重。

5 流程概述

命令行的主要流程为：[载入词库]->[读取网页]->[提取关键信息]->[输出信息结果]->[中文分词]->[构建倒排文档]->[读取查询字符串]->[对查询字符串分词]->[检索]->[结果排序]。

部分步骤采用了多线程。GUI 部分采用动态链接库调用倒排文档等的 API，除了上述步骤外，JavaScript 还会包含以下步骤：[根据关键词位置进行文档摘要]->[高亮关键词]->[绘制进度条等信息]。

6 输入输出及操作相关说明

query.exe 为命令行程序，**直接运行即可**。程序会自动搜寻相对路径 input。遍历其下的所有文件作为 html 解析，搜索 dictionary.dic 和 stop-words.dic 载入词典。**请勿在 input 目录下放置其他文件**。

gui.bat 为图形程序的启动脚本，起到设置工作目录的作用，**直接运行即可**，不依赖于命令行程序。不首先进入主界面，等待进度条加载完毕。点击“Sunlab Search”按钮或是在输入框内回车即可发起搜索。点击“I'am Feeling Unlucky”可直接跳转到第一个匹配的页面。在结果页面，点击 logo 可回到主界面，点击搜索到的条目会直接打开网页。Ctrl-Shift-I 会打开控制台，Ctrl-R 或是 F5 会重新载入页面。

7 实验结果

符合预期。

8 功能亮点说明

~~0. 代码没啥注释。QAQ 上次扣过了，求只扣一次就好~~

1. 性能好，垃圾 Windows 太慢了，要 1.5s，我 Linux，0.6s 就可以。
2. 搜索页面做得到，高亮到每个词，还支持英文搜索，摘要也摘在了关键词附近。
3. 错误处理做得好。甚至如果 html 标签不匹配都可以适当的报错，给出 html 文件的路径、行号等等。
4. 我……编不下去了

9 实验体会

作业好简单啊，不够达到训练的目标，求加量，求加难度。