**搜索迷你项目任务时间表**

1. 项目需求

开发一个搜索引擎迷你项目，涉及分词，建索引，网页去重，计算相似度，文本聚类，前台页面展示，多进程编程，网络编程，守护进程编写，makefile编写，工程组织等各方面内容。

目的是让大家了解一个小工程的各个方面

1. 开发环境

Linux操作系统下，C++开发

1. 项目目录组织

目录结构

src

include

lib

bin

log

conf

data

Makefile

1. 项目模块划分
2. 组织项目结构
3. 建立索引
4. 分词（这里主要是学习如何使用库文件）
5. 网页去重
6. 相似度计算
7. 文本聚类 -- 备选
8. 网络IO库的建立及使用
9. 项目模块需求描述
10. 组织项目结构，编写基本的makefile，配置文件, 设计系统的框架

组织目录结构，编写配置文件，编写基本的makefile ----7月29日上午

配置文件名字：**search.conf**

Conf文件格式

<Data>

Index\_file=../data/index.dat

Page\_file=../data/page.lib

Query\_file=../data/query.lib

Stopword\_file=../data/stoplist.txt

…….

<Socket>

Port=54321

设计系统的框架，设计项目中需要的类文件 ----7月29日下午

1. 建立索引 7月30日

逐个读取query.lib 中的每一个词，然后读该词在ripepage.lib中的偏移位置，保存该位置到index.lib文件中，一共两个字段，分别为查询词，查询词在网页库文件中的偏移，具体格式为：

Keyword tab offset

注意，这里建立索引时，需要分析网页库中的文件格式，即根据网页库格式，找出每个查询词对应稳定列表的开始

1. 分词 7月31日

这里主要是学习在makefile中如何连接一个库文件，分词库具体的使用在模块3中。

分词库的使用：

包含头文件：

#include "SegInterface\_Use.h"

#include "PosInterface\_Use.h"

#include "codetransformer.h"

#include "JTCharCodeConvert.h"

Demo程序文件：testelus2.cpp， 在elus\_interface目录下

1. 网页去重 8月1日 – 8月3日

在该模块，(包含去停用词操作)

每次读取query.lib中的一个词，执行以下操作。

到index.lib中找到该词在网页库中的偏移位置，依次读出该词对应的所有文件，对于这些文件，进行网页去重，去重的方法有以下两种（两种方法大家都实现一下）：

8月1日-8月2日

1. 为每个网页提取网页中的特征码，特征码为该网页中词频（就是一个词在该网页中出现的次数）最高的10个词----top10词；对于每两篇网页，比较top10词的交集，如果交集大于6个，认为它们是互为重复的网页。在计算top词的时候，就需要使用分词程序，即将一篇网页分词，然后统计词频。

注意：这里取top10词的时候要做停用词过滤，停用词即常见的词，如：的，啊，个等词，会提供一个停用词表给你们。

8月3日

1. 为每个网页提取网页中的特征码，特征码为：提取网页中每个逗号前后的5个字节，这样每个网页可以提取出若干的字符串，将这些字符串顺序链接形成一个特征字符串S，对于每两篇网页，计算特征字符串S1和特征字符串S2之间的最长公共子序列LCS，如果len(LCS)/min(len(S1), len(S2))>0.6, 则认为重复。

(每个文件是一个对象的实例，对象可以包括title，content，docid，在网页库中的offset信息等成员变量，可以选择vector或者list存储，自己测试下用什么数据结构效果好)

这里的输出应该是一个词对应了一个文档集合，该集合里每个文档都是不重复的。

8月4号，没做完的同学补一下，做完的同学，继续往下或者休息

1. 计算文本相似度 8月5日- 8月7日

统计出网页库中出现过的所有的词，得到一个词集S。统计S中每个词w在多少个文档中出现过，这个信息称为该词W的文档频率df，统计处词w在每个文档中出现的词数，这个信息称为词频tf，例如，对于文档d1，w在d1中出现了x次，则称在d1中w的词频tf为x。 对于文档d1，计算出，每个词的tf值，然后计算每个词的特征权重，计算规则如下：

对于词w，在文档d1中的权重为，tf d1\*(1/dfw)

在文档d2中的权重为，tf d2\*(1/dfw)

这样，对与一个文档中的每一个词，都可以计算出这样的权重，例如对于文档d1，可以得到一个权重序列

W1,w2,w3……..wn, 然后需要归一化，即w1=w1/(w1+w2+…….wn)

即每个文档表达成一个特征向量，计算每两篇文档间的相似度，使用向量相乘，成为余弦相似度。

1. 查询模块

返回网页信息。

1. 前台页面