搜索引擎大作业报告

钱迪晨　计 35　2013011402

温和　计 35　2013011407

2016 年 6 月 17 日

1 爬虫

爬虫使用的是 python 自己写的爬虫。我们使用 eventlet 协程来控制 IO，而不使用多线程进行 IO 控制，大大降低了 CPU 的负荷。实际测试中，可以同时支持 300 个 HTTP 请求的 IO，速度非常快，如果是多线程的话只能达到大概 50 个。当然最后爬数据的时候，只开了 10 个协程。但是速度非常快，大概 8 个小时就爬了 20 万网页。

数据存储在本地，每 1 万个数据存储一个文件夹。每个数据有原始数据以及 json 格式的描述文件，包含一些信息。

爬虫也做了中断保存的功能，可以随时中断，然后继续上一次的任务队列。同时爬虫也做了简单的命令行指令，可以随时加入种子网页，以及保存退出。

2 文件解析

文本解析使用 Apache 的 Tika（对于 office 文档和 pdf）和 python 中的 BeautifulSoup4（对于 html）完成。对于 office 文档，直接使用 Tika 提取文件中的文本和作者等信息；对于 html，使用 bs4 提取 body 中的文本。对于爬虫得到的 20 万个结果，共处理了 10 小时左右。

对于 html 文件，我们还提取了一张 body 中的图片（去除明显的 banner、logo 等图片），作为显示搜索结果时的预览图。

3 lucene

lucene 使用了最新的 6.0. 分词器使用的是 smartCNAnalayzer，效果很不错。原来作业里面的 lucene 版本太老旧。

建立索引的时候，将 pagerank 的值乘到每个 document 的每个 field 里面去。（lucene6.0 不支持 document boost）。同时在建立的索引的时候，将一些信息也保存进去，比如这

个 document 的原网址，网址的图片，这些用来渲染的必须信息。由于 pagerank 在量级上差距过大，我们取系数为$1.5 * (log_2(pr) + 22)$。

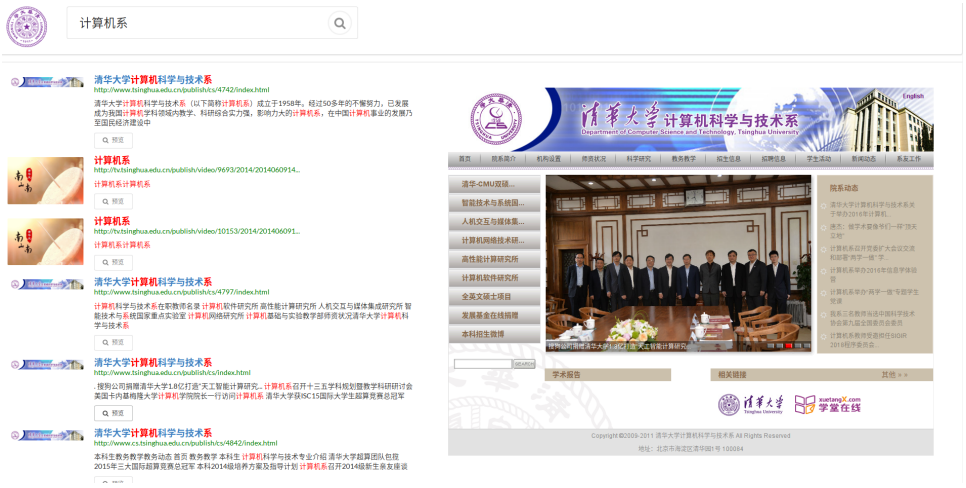
搜索的时候使用的 query 是 MultiFieldQueryParser 用来构建询问。关于评分函数，我们使用了标题，文本和作者，权重分别为 1，0.01 和 0.5，对于 html 还有额外的 h1、h2、h3 和 a，权重分别为 0.05,0.03 和 0.02,。

4 前端

网站前端我们使用的是 semantic2.1，以及 jquery。

搜索初始页只有一个标题和搜索框，非常简洁。

页面最上面是一个搜索栏，显示当前的搜索关键词。结果页面分成两部分，左半边是搜索的结果以及页面的切换。右半边是页面的浏览功能，这个功能如果是有上下滑动的触摸板的话效果非常好。效果如下图：



5 代码详细

在 src 文件夹中包含三部分。

crawler 是由 python 写的爬虫部分。

script 是由 python 写的进行数据处理的部分。

webserver 是使用 lucene6.0，java 写的搜索器。