

北京大学中文系网站站内索引开发报告

Java 程序设计大作业

一. 用户分析

1. 中文系网站定位

北京大学中文系网站是北京大学中文系的官方主页,主要反映的是中文系师生的学习工作情况,同时又有中文系主要部门和大多数老师的职能介绍。在中文系网站首页,有以下几个板块:本系简介、动态新闻、本系公告、教工园地、学生园地、图书分馆、精品课程、教学科研、主办期刊、英文主页、中文论坛、网站管理。这些板块的内容大多是粗略地介绍中文系目前的情况,教职工、学生的工作学习生活,中文系官方的公告等。

中文系网站主要面向的对象是北大中文系学生、有意向了解中文系的其他院系学生或校外人士。对于北大中文系学生,有价值的主要信息在于本系公告,他们能通过公告栏得知目前中文系的一些政策,对于非中文系的学生,简介和新闻则比较重要,他们会想要了解中文系大致的介绍,中文系一般的活动是怎样的,这些可以通过站内的这些内容进行查找。

2. 检索系统的定位

我们做的检索系统是北大中文系的站内检索,搜索的所有内容都在站内,所以使用北大中文系网站的用户都是检索系统的用户。对于用户来说,有价值的内容主要是每一篇文章的标题和内容,在每一个网页中,只有这两部分是最有价值的,而其他的链接等,对用户而言,没有检索意义。如在上图中,导航栏的链接内容对于用户而言是没有意义的,有意义的是标题“北京大学中文系简介”,“日期”,“信息来源”和下面的正文内容。

这些有检索价值的信息决定了我们在设计网页解析时要解析的内容,查看源代码可以发现,图 1 的内容是我们并不需要的。这些内容是导航中的内容,所以我们并不把它们纳入检索范围。图 2 中的主要内容才是我们检索的对象。在后续的过程中,我们将这些内容爬下来并进行存储,然后利用工具进行切词、建立索引、检索。总而言之,这些有着实在意义的内容和标题,才是我们在建立信息系统过程中的处理对象。

图 1 导航菜单源代码

图 2 正文内容源代码

图 3 北京大学中文系简介

上面已经说到,使用我们的检索系统的用户是使用北大中文系网站的所有用户。在分析

用户定位前，我们可以将用户群体分为北大中文系学生和非北大中文系用户。

1.1 北大中文系学生

对于北大中文系的学生而言，他们对本系较为了解，对本系的大致情况并没有很大的兴趣，主要的信息需求是一些确切的信息。比如他们会想要了解中文系寒假值班的具体情况，或者关于本系某一活动的具体信息。这些用户的信息需求是明确的，这种信息需求通过检索行为表达出来，就是较为明确的查询行为，且这类用户的检索水平较高，专业知识较高。所以我们提供了标题检索和全文检索，标题检索可以帮助用户可以快速准确查到特定的信息（因为官方网站的标题往往直接反映主题内容），而全文检索则有助于帮助用户定位内容中的确切信息。

2.1 非北大中文系用户

非北大中文系的用户往往会比较想要了解一些关于北大中文系的介绍，或者是关于部分老师的介绍，他们的信息需求较为宽泛且并不是很明确，对于检索结果也不要求精确定位，他们的潜在信息需求往往会在浏览网页时渐渐明确。比如他们会直接搜索“北大中文系”或者是搜索某位老师的名字比如“孔庆东”。对这类用户而言，全文检索比较适合他们，且他们较为宽泛的检索词往往会导致较多的检索结果。我们通过相关性的排序，保证相关性高的检索结果排在最前面，使用户在浏览查询结果时，能够获得相关性最高的信息，满足自身的信息需求。

二． 工具选择

1. 爬虫

3.1 Nutch

总体上 Nutch 可以分为 2 个部分：抓取部分和搜索部分。抓取程序抓取页面并把抓取回来的数据做成反向索引，搜索程序则对反向索引搜索回答用户的请求。抓取程序和搜索程序的接口是索引，两者都使用索引中的字段。抓取程序和搜索程序可以分别位于不同的机器上。

抓取程序是被 Nutch 的抓取工具驱动的。这是一组工具，用来建立和维护几个不同的数据结构：web database，a set of segments，and the index。

The web database，或者 WebDB，是一个特殊存储数据结构，用来映像被抓取网站数

据的结构和属性的集合。WebDB 用来存储从抓取开始（包括重新抓取）的所有网站结构数据和属性。WebDB 只是被 抓取程序使用，搜索程序并不使用它。

Segment，是网页的集合，并且它被索引。Segment 的 Fetchlist 是抓取程序使用的 url 列表，它是从 WebDB 中生成的。Fetcher 的输出数据是从 fetchlist 中抓取的网页。

The index。索引库是反向索引所有系统中被抓取的页面，它并不直接从页面反向索引产生，而是合并很多小的 segment 的索引产生的。Nutch 使用 Lucene 来建立索引，因此所有 Lucene 相关的工具 API 都用来建立索引库。

抓取是一个循环的过程：抓取工具从 WebDB 中生成了一个 fetchlist 集合；抽取工具根据 fetchlist 从网络上下载网页内容；工具程序根据抽取工具发现的新链接更新 WebDB；然后再生成新的 fetchlist；周而复始。一般情况下，我们不需要接触底层的工具，只要从头执行程序就可以了。

在进行了相应的系统配置后，将会得到如图 4 的文件夹，并且将 Nutch.war 放到 Tomcat 中，进行一定的配置，运行界面。



 crawldb	2015/1/15 22:28	文件夹
 index	2015/1/15 22:37	文件夹
 indexes	2015/1/15 22:20	文件夹
 linkdb	2015/1/15 22:19	文件夹
 segments	2015/1/15 22:19	文件夹

图 4 nutch 爬取收文件夹



图 5 nutch 检索结果界面 1



图 6 nutch 检索页面 2

我们发现已经实现了一定程度上的站内检索图 5 图 6，但是我们最后放弃了这样的做法，一方面是想要自己动手走一遍流程，另外一方面认为这样的检索效果并不令人满意。当然可以对其进行进一步的定制，例如使用更好的切词，在深度和广度上进行进一步的设置等。

4.1 不用 Nutch 转用 Heritrix 的原因

Nutch 虽然不必我们接触底层的工具，但是处理起来也是非常繁琐，所以我们开始考虑换一种爬虫工具，即 Heritrix。

总体来说 Heritrix 网络蜘蛛的功能更为强大，而 Nutch 更好地支持搜索引擎（与 Lucene 紧密结合）。两者特点对比如下：

Nutch 是一个搜索引擎框架，

	Nutch	Heritrix
功能	获取保存索引内容	下载，爬取
类型	可索引部分	各种类型
任务	合并索引	任务管理
管理	命令行	Web 定制界面
控制	参数少	灵活

Heritrix 中有几个关键模块：下载控制器 CrawlController，这是总近控部分，以主

2. 网页解析

随意打开中文系网站的一个页面，如动态新闻里面的“北大中文系海外名家讲座系列—郭实腊，英国圣教书会出版物与传教士中文小说”，会发现中文系的网站结构设计地非常简单，除去上方各种不同的链接之外，就只剩位于中部的主要内容了，因此网页解析比较简单。

在网页解析这一步骤中，我们也是使用的开源工具 HTMLParser，这是一个小巧的网页解析工具，只是相关文档较少，很多功能需要自己摸索。通过查阅资料，我们了解到，HTMLParser 的核心模块 `org.htmlparser.Parser` 类，这个类完成了对于 HTML 页面的分析工作。HTMLParser 遍历网页内容后，可以依据 HTML 文件以树状结构组织数据的特点，得到以树（森林）结构保存的结果。

在解析网页这一过程中，我们还使用了 Tika 框架进行网页解析，Apache Tika 可以利用现有的解析类库，从不同格式的文档中（例如 HTML，PDF，Doc），侦测和提取出元数据和结构化内容。它可以侦测文档的类型，字符编码，语言，等其他现有文档的属性；提取结构化的文字内容。如图：

在最终的比较中，我们认为 HTMLParser 更好用，但舍弃了 Tika 框架，可能还是觉得更加适合我们的任务本身，当然如果在后期对于网站中的 doc 文件也建立索引，使用 Tika 显然也是极好的。并且有利于我们进一步掌握更多的文件的解析方法。

3. 切词工具

分词准确性对搜索引擎来说十分重要，对于搜索引擎来说，分词的准确性和速度，二者都需要达到很高的要求。但在我们的检索系统中，并不是动态的，所以在一定程度上，对于准确性的要求更高。切词建索引方面，我们结合课程和老师给的例子，使用了老师推荐的 IKAnalyzer 作为切词的工具，并进行相关业务的定制，以便更好的满足我们的需求。主要包括对于停用词表的定制等。

三． 爬建查索

1. 爬

爬虫，如前文提到的，我们首先使用了 Nutch 作为爬虫工具，实现了一定程度的检索，而在 Nutch 需要 Linux 环境支持，所以安装了 Cygwin64 Terminal，然后定制了

<http://chinese.pku.edu.cn> , 之后进行 `bin/nutch crawl urls/url.txt -dir crawled -depth 4 -threads 5 -topN 1000 >&logs/log1.log` 即深度为 4 线程为 5 的爬取, 利用 1.2 版本自带的检索工具, 进行了检索, 页面如图 5 图 6。这个过程中, 我们发现, 可以使用 solr 工具进行检索, 并且在 Nutch1.2 以上的版本中, 不再携带检索的页面, 而是将业务分给 solr。Solr 是一个独立的企业级搜索应用服务器, 它对外提供类似于 Web-service 的 API 接口。

但是, 这些底层的封装, 不利于我们进一步理解检索系统, 因此, 我们决定自己动手做一个检索系统。而爬取是我们的第一步。我们的想法是将网页整体镜像爬下来, 然后再考虑对镜像文件进行分析, 建立索引。

我们尝试修改老师的爬虫程序, 已达到自己的定制过程, 但是限于精力和时间, 并且对于我们的系统而言, 重心是索引本身, 另外, 以 heritrix 为代表的开源爬虫工具比较成熟, 老师上课鼓励使用开源工具, 因此, 我们使用了 heritrix 工具, 用镜像的方式保存到本地。安装过程比较麻烦, 已打包为 H1 项目, 启动后, 将会进入 web 的定制页面。其中需要指出的是, 这个过程中对于 Jdk 的版本有一定的限制。需要 JDK1.5 以上版本。

2. 建

将爬虫的信息导入到数据库中, 以数据库为中间载体, 建立全文索引, 而如何将如图所示的大量网站中提取有价值的 htm 文件, 并对文件进行解析是建立索引的准备工作。

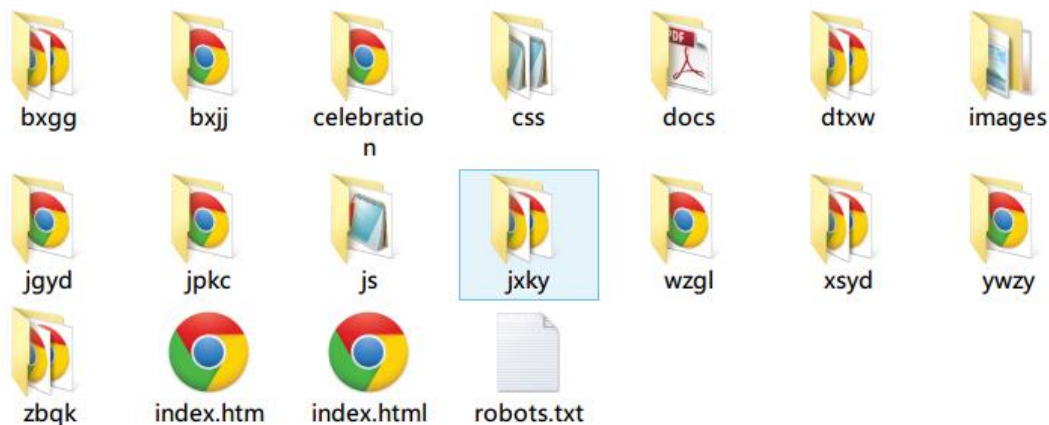


图 9 中文系网站结构

名称	修改日期	类型	大小
 20140404101612047736.doc	2015/1/17 16:55	Microsoft Word ...	89 KB
 20140404101801653828.doc	2015/1/17 16:55	Microsoft Word ...	39 KB
 20140404102050471932.doc	2015/1/17 16:56	Microsoft Word ...	66 KB
 20140404102318096040.doc	2015/1/17 16:55	Microsoft Word ...	40 KB
 20140404102555016153.doc	2015/1/17 16:55	Microsoft Word ...	40 KB

图 10 doc 文件夹中的部分截图

首先,根据用户需求,爬虫爬下来的网站中,存在很多文件,包括 js,css,甚至有许多 word 文件,是该网站的文章,而我们的定位里,将这部分信息暂时并不包含在检索中,当然,如果进一步提高检索的检全率,这部分也是可以使用相应的工具进行解析的。因此,锁定网站中的 htm 文件,并指明为类似 4375.htm、4539.htm 等数字开头的文件,而不是 index1.htm 等目录文件。

具体代码为用一个队列进行本地的广度优先的变量,通过正则表达式,判断其是否为满足要求的文件,对于文件进行解析。

```
for (File i:filelist)
{
    if (i.isDirectory()) //如果是目录
    {
        deque.add(fstr+"/"+i.getName());
    }
    if (i.isFile()) //如果是文件,只要判断是都以htm结尾,并不是Index开始的文件进行解析
    {
        String name=i.getName();
        if (name.endsWith(".htm") && (!name.startsWith("index")))
        {
            String tmp=fstr+"/"+name;
            try {
                d2 one=new d2(tmp);
            } catch (FileNotFoundException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            } catch (ParserException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            } catch (SQLException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
    }
}
```

图 11 队列解析文件部分代码

而文件的解析,是根据网站结构而言的,在前文已经提到,便不再赘述,主要通过

```
NodeFilter filter1=new AndFilter(new TagNameFilter("div"),new
HasAttributeFilter("class","content"));
NodeFilter filter2= new AndFilter(new TagNameFilter("p"),new
HasAttributeFilter("class","detail-p"));
NodeFilter filter3= new CssSelectorNodeFilter("h2");
NodeFilter filter4=new AndFilter(new TagNameFilter("div"),new
HasAttributeFilter("class","detail-size"));
```

通过对于相应的 div 或者元素选择器,建立 filter 类,而将这些解析后的代码通过本地文件或数据库保存起来,而在本程序中使用的是 Mysql 数据库进行保存,便于进一步的开发和利用,值得一提的是,在提取日期时,使用了正则表达式,对于非数字的文本进行了剔除,最后导入数据库中,如图所示。

玉罗涛刘倩刘琰赵妮周雅娜张丹星冯

例如对于漆永祥，切词将其进行了切开，或许，这并不是最好的处理方式。



```
Field id2=new
Field("id",id1,Field.Store.YES,Field.Index.NO,Field.TermVector.NO);
doc.add(id2);
Field title2=new
Field("title",title,Field.Store.YES,Field.Index.ANALYZED,
Field.TermVector.WITH_POSITIONS_OFFSETS);
doc.add(title2);
Field URL2=new
Field("URL",URL,Field.Store.YES,Field.Index.NO,Field.TermVector.NO);
doc.add(URL2);
Field date2=new
Field("date",date,Field.Store.YES,Field.Index.NO,Field.TermVector.NO);
doc.add(date2);
Field content2=new
Field("content",content,Field.Store.YES,Field.Index.ANALYZED,Field.Term
Vector.YES);
doc.add(content2);
```

对于标题和全文需要提供检索，因此对于词向量进行了处理，以便后面的高亮的工作能够正常进行。而在 buildtheindex 包中，我们写了一段小代码进行检验，结果如下。

```
加载扩展停止词典：stopword.dic
找到：403 个结果！
上海中文系系友举行“纪念北大中文系100年”茶话会
249
/dtxw/4562.htm
0.29887018
“中国作家北大行”之五——王蒙：现当代文学中的几个重要问题
65
/dtxw/4057.htm
0.2883608
第五届北京大学中文系王默人小说创作奖获奖名单
247
/dtxw/4559.htm
0.2883608
中文系2012年学科发展战略研讨会顺利召开
294
/dtxw/4620.htm
0.2676212
北大中文系开设“胡适人文讲座”
163
/dtxw/4426.htm
0.2648763
```

图 13 buildtheindex 运行代码



北京大学
中國語言文學系
Department of Chinese Language and Literature

北京大学中文系站内检索



北京大学信息管理系
DEPARTMENT OF INFORMATION MANAGEMENT

信以通达，管则治世

Search

☐ 标题 ☒ 全文

Copyright - 黄俊杰,倪少康,童刘奕,冀伟浩,牛嘉诚

Java课程作业



北京大学
中國語言文學系
Department of Chinese Language and Literature

北京大学中文系站内检索

Search



标题



全文

北大中文系找到：415 个结果！

上海中文系校友举行“纪念北大中文系100年”茶话会

chinese.pku.edu.cn/dtxw/4562.htm

北大上海校友网于2010年8月29日，在已有百年历史的上海最著名会所“张园”举行了“纪念北大中文系100年”茶话会，庆祝北大中文系即将到来的百年华诞（见下图）。北大上海校友网（<http://bbs.pkushanghai.cn>）建立于2005年，是一个由现在上海工作和

北大中文系系庆活动现场

chinese.pku.edu.cn/dtxw/4603.htm

来源：新浪教育 时间：2010年10月25日 2010年10月23日上午，北京大学中国语言文学系建系100周年庆祝大会在百周年纪念讲堂隆重举行。以下为系庆当天活动现场照片。北大百年大讲堂举办隆重的庆祝活动 北大中文系所在地：五院 来自海内外的新老系友欢聚一堂

北京大学中文系建系百年

chinese.pku.edu.cn/dtxw/4602.htm

来源：光明日报 时间：2010年10月24日 10月23日，“我们不能设想，如果没有中文系，北大是个什么样子。”中央文史馆馆长袁行霈教授说。北京大学中国语言文学系今天举行建系100周年庆祝大会，数百名校友重聚燕园，回眸北大中文系和我国大学中文教育这一百年的风云际会、沧海桑田。百年校庆都

北大有群旁听生——兼贺北大中文系诞辰100周年

chinese.pku.edu.cn/dtxw/4471.htm

来源：新疆日报网 2010年5月4日 （新疆日报网讯）今年3月31日，是北大中文系建系100周年。百年华诞，百年辉煌，北大中文系在轰轰烈烈举办各式各样的庆祝活动的同时，我也不禁想起了一群曾在北大旁听的“偷听生”。北大旁听生，有人称他们为“偷听生”“蹭客”

北大中文系庆建系100周年 系主任否认生存困难

四. 系统评价

我们主要通过系统自身的效率和与其他检索系统对比进行系统的评价,同时,我们还进行进一步的反思,为我们系统未来的改进提出了设想。

1. 检全率与检准率

检索系统的检全率和检准率是评价信息系统的重要指标。

检全率是指系统实施检索时检出的于某一检索提问相关的信息资源量与检索系统中与该提问相关的实有信息资源总量之比,可以表示为

$$\text{检全率} = \text{检出相关信息资源量} \div \text{系统相关信息资源量} \times 100\%$$

以检索“孔庆东”为例,通过百度检索出中文系网站内的资源是 10 条,通过谷歌检索出中文系网站内的相关资源是 13 条,通过我们自己的检索系统检出 12 条,在检全率这一指标上,谷歌比我们的检索系统高,而百度则比我们的检索系统低。百度没有查到“孔庆东赴台”的新闻,而谷歌比我们多查到一条资源,如图所示。究其原因,是谷歌比我们多做了目录页的索引。而在我们的定位分析中,我们认为并不需要做目录页。这就造成了我们和谷歌检索的结果的差异。

Page 2 of 13 results (0.16 seconds)

11 - 北京大学中文系

chinese.pku.edu.cn/dtxw/index10.htm ▾ Translate this page

... 未名大讲堂: 张颐武教授对话王小丫; [2010-04-25]王岳川: 警惕“反经典”和“伪经典”; [2010-04-25]陈跃红、孔庆东成都授课; [2010-04-25]陈平原: 何为大师?

图 14 Google 多出的检索结果

检准率是指系统实施检索时检出的与某一提问相关的信息资源量与检出的信息资源总量之比,可以表示为

$$\text{检准率} = \text{检出相关信息资源量} \div \text{检出信息资源总量} \times 100\%$$

同样以检索“孔庆东”为例,在这个例子中,百度、谷歌和我们自己的检索系统查到的所有信息资源量都与孔庆东有关,也就是说,检准率都是 100%,是比较高的,当然这和信息资源本身容量较小也有关。

检全率与检准率两者本身也具有一定的关系,在大多数情况下,检全率与检准率具有一定的互逆关系,这在大多数系统中都通过排序的方式进行,我们的系统也不例外。我们使用的是向量空间模型的算法,通过“tf*idf”的计算方式,根据检索词和检索结果的匹配程度进行排序,保证相关性较高的检索结果排在前面,便于检索结果的展示,提高检全率和检准率。

仍以检索“孔庆东”为例,下图是百度、谷歌和我们自己的检索系统给出的排序,值得一提的是,我们的检索系统通过标题查询的排序与谷歌是一样的,和百度有一定差别,且百度比

我们的检索结果少了一个，从这个结果而言，我们的结果接近谷歌检索系统，优于百度的检索效果。

百度为您找到相关结果约10个

[孔庆东 - 北京大学中文系](#)

孔庆东日期: 2013-12-31 信息来源:中文系个人简介 孔庆东,北京大学中文系教授 联系方式 北京大学中文系100871 专著与论集《超越雅俗》,《1921:谁主沉浮》,《...
[chinese.pku.edu.cn/jgy... 2014-3-18](#) - 百度快照 - 94%好评

[陈跃红、孔庆东成都授课 - 北京大学中文系](#)

2010年4月25日 - 本报讯(黄鹏程记者周波)昨天,北京大学中文系教授陈跃红、孔庆东以及物理系教授舒幼生出现在成都七中嘉祥外国语学校,在拟持续两个月的“嘉祥导师系列...
[chinese.pku.edu.cn/dtx... 2010-04-25](#) - 百度快照 - 94%好评

[北京大学中文系](#)

2014年10月24日 - 陈平原、吴晓东、孔庆东、高远东 陈晓明、张颐武、李杨、韩毓海、计璧瑞 050108 比较文学与世界文学 1. 比较诗学 2. 文化研究 比较文学及中外...
[chinese.pku.edu.cn/bxg... 2014-10-24](#) - 百度快照 - 94%好评

[北京大学中文系](#)

博士研究生:方锡德、解志熙、孔庆东、李书磊、旷新年、刘为民、李惠彬、范智红、李今、申正浩(韩)、田炳锡(韩) 硕士研究生:吴福辉、凌宇、江锡铨、王友琴...
[chinese.pku.edu.cn/jgy... 2013-12-30](#) - 百度快照 - 94%好评

Figure 1 百度查询结果排序

[孔庆东 - 北京大学中文系](#)

[chinese.pku.edu.cn/jgyd/xdwxjys/4118.htm](#) - Translate this page

Dec 31, 2013 - 个人简介孔庆东,北京大学中文系教授联系方式北京大学中文系100871 专著与论集《超越雅俗》,《1921:谁主沉浮》,《金庸评传》,《中国现代文学 ...

[陈跃红、孔庆东成都授课 - 北京大学中文系](#)

[chinese.pku.edu.cn/dtxw/4423.htm](#) - Translate this page

Apr 25, 2010 - 来源:四川新闻网-成都日报(成都) 2010年04月25日 本报讯(黄鹏程记者周波)昨天,北京大学中文系教授陈跃红、孔庆东以及物理系教授舒幼生出现 ...

[孔庆东赴台参加第二届“中华文化快车”活动 - 北京大学中文系](#)

[chinese.pku.edu.cn/dtxw/4349.htm](#) - Translate this page

孔庆东赴台参加第二届“中华文化快车”活动. 日期: 2009-09-28 信息来源: 中文系. 来源: 中国台湾网2009-09-26 台湾网9月26台北消息第二届中华文化快车两岸 ...

Figure 2 谷歌查询结果排序



Figure 3 我们的检索系统排序（标题检索）

由于我们的检索系统是一个小型的检索系统，信息资源量非常小，所以在检全率和检准率方面，我们比较看重的是检全率，在信息资源量不多的情况下，如何使检索出来的结果尽可能多，尽可能包含所有相关信息资源，是我们比较关心的问题。通过一系列的测试，我们认为我们的检索系统的检全率还是比较高的。

2. 检索速度与易用性

评价信息系统的另一个主要方面是检索速度与易用性。

信息系统的检索速度是指用户实施检索时获得检索结果花费的时间。通过测试，可能是由于检索的资源量较少，且我们使用倒排档的索引方式，所以我们的检索速度比较快，能够及时反馈给用户他们的检索结果。

信息系统的易用性对用户而言是指检索系统是否易于使用，一般会具体到操作、界面等方面，也包括容易获取信息的程度。在我们的检索系统中，操作非常简单，检索方式只有两种——标题检索和全文检索，用户只要输入想要检索的内容，系统会自动将用户指令切词，然后进行匹配检索，迅速给出结果；界面非常友好，我们的界面同时体现了中文系深厚的底蕴和信息管理系的职责使命，对于用户而言，这样的检索系统用着会比较舒服；查询结果直接点击可以进入原网站，信息获取非常容易。

3. 改进空间

在检索系统的构建过程中，我们遇到了一系列问题，最终总算完成了作业。检索系统的很多地方都是使用开源软件，便于我们的修改和完善。在我们的检索系统中，还存在一定的不足。我们建立索引的项目并没有做到完全覆盖。如文章中的时间信息我们没有单独的提取出来建立索引，这就限制了结果展示更丰富的维度。此外，网站中以附件形式存在的 Doc 文

档我们采取了存储的方式，也没有解析、建立索引。未来可以对这些文件中的信息也覆盖来提供检索。

4. 感想

我们整体的项目框架采取了 MVC 结构，这样的结构使得我们在项目进行时的思路更为清晰。后期随着合作的进行和调整，我们将 M(Model)和 C(Controller)合并到了一个 JavaBean 之中，将 V(View)单独设置在了 JSP 中。

在系统的建设过程中，通过团队合作，交流，进一步加深了我们的团队合作精神，另外在这个过程中，我们采用了很多的开源工具，一定程度上，加深了我们的难度，但我们也因此收获很多，了解了许多成熟的技术。作为信息管理系的学生，我们发现自己需要进一步提高的地方还有很多，不断的学习的过程将会让我们在信息的处理的能力上有更多的提高。这也是我们认为在整个项目中最为重要的部分。