

**Geeking搜索引擎项目报告**

团队名称：GeeK

团队成员：

陈新荃 2014E8018461048

高妍

林裕杰

肖卡飞

2014年12月14日

目录

[**一．项目总体介绍** 3](#_Toc405806038)

[**二．设计方案** 3](#_Toc405806039)

[1. 网页爬虫 3](#_Toc405806040)

[1.1 爬取策略 3](#_Toc405806041)

[1.2 具体实现 3](#_Toc405806042)

[2. 索引构建 3](#_Toc405806043)

[2.1 数据结构 3](#_Toc405806044)

[2.2 构建流程 4](#_Toc405806045)

[3. 检索策略 4](#_Toc405806048)

[3.1 检索流程 4](#_Toc405806049)

[3.2 结果排序 4](#_Toc405806050)

[3.3 结果聚类 4](#_Toc405806051)

[4. 前端处理 5](#_Toc405806052)

[4.1 页面元素 5](#_Toc405806053)

[4.2 自动补齐 5](#_Toc405806054)

[4.3 搜索词推荐 5](#_Toc405806055)

[4.4 摘要快照及高亮 5](#_Toc405806056)

[**三．测试与评估** 5](#_Toc405806057)

[1. 测试环境 6](#_Toc405806058)

[2. 测试结果 6](#_Toc405806059)

[2.1 功能测试 6](#_Toc405806060)

[2.2 性能测试 6](#_Toc405806061)

[**四．创新点** 6](#_Toc405806062)

[1. 标题权重 6](#_Toc405806063)

[2. 中间字自动补全 6](#_Toc405806064)

[3. 【还有什么尽管添加】 6](#_Toc405806065)

[**五．经验与总结** 7](#_Toc405806066)

[1. 个人经验总结 7](#_Toc405806067)

[2. 组长总结 7](#_Toc405806068)

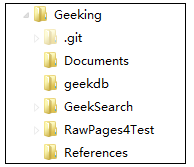
# **一．项目总体介绍**

Geeking，是一款体育新闻搜索引擎，由GeeK团队开发。Geeking能够爬取各大门户网站的体育新闻（目前支持搜狐、腾讯、网易、MSN），并建立成倒排索引，实现体育新闻搜索。其功能包括：相关度排序、相似新闻聚类、搜索词推荐、网页摘要、关键词高亮、网页快照等。

Geeking项目由Java语言编写，总代码量约为3750行（包括java、jsp文件以及空行）。该项目的实现除了团队编写的核心内容之外，还采用了htmlparser（网页过滤）、ansj\_seg（ict.中文分词）等第三方工具辅助开发。

团队使用GitHub作为协同开发工具，目前项目版本为0.2，总共发生了超过180次代码提交。项目地址为：<https://github.com/UCAS-GeeK/Geeking>。

项目目录结构如下：



Documents：团队开发过程中积累的开发文档，包括例会日志、协同开发教程、设计文档等等。

geekdb：团队开发过程中使用的小型数据库。

GeekSearch：项目工程，开发平台为eclipse。

RawPages4Test：测试用的html网页。

References：参考文档，包括代码规范等。

本报告后续内容将根据Geeking项目工程GeekSearch来展开。

# **二．设计方案**

GeekSearch工程主要模块结构如下：

//工程截图介绍

下面详细介绍各个模块的设计与实现。

## 1. 网页爬虫

### 1.1 爬取策略

🡪咖啡

### 1.2 具体实现

🡪咖啡

第三方jar包等

## 2. 索引构建

🡪裕杰

总体介绍

### 2.1 数据结构

🡪裕杰

数据库、索引在内存中的结构等

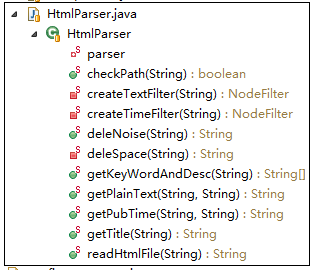
### 2.2 构建流程

🡪裕杰

### 2.3 网页过滤

本项目的网页过滤部分采用开源第三方工具htmlparser（http://htmlparser.sou

rceforge.net/），并结合了正则表达式过滤等方法，加以封装，以实现项目需求。封装后的代码结构如下：



该类主要实现了对网页信息的抽取，包括title、description、keywords、public time以及网页正文等。目前可实现对搜狐、网易、腾讯、MSN四大门户网站的新闻网页过滤。其中四大门户网站网页中的title、description和keywords内容均可以通过相同的网页标签定位并获取。但是对于public time，四大网站的标签均不同，并且对于同一个网站，其不同年份的public time标签也都不同，因此此处并未采用htmlparser来抽取，而是使用了正则表达式，具体实现如下：

### 2.4 文本分词

项目中的分词模块使用了Ansj（https://github.com/NLPchina/ansj\_seg）第三方开源中文分词工具。Ansj作者孙健，其实现是基于中科院的 ictclas 中文分词算法，比其他常用的开源分词工具（如mmseg4j）的分词准确率更高。

**分词方法**

Ansj主要分词方法有：基本分词（BaseAnalysis）、精准分词（ToAnalysis）、nlp分词（NlpAnalysis）、面向索引的分词（IndexAnalysis）。四种方法的对比如下表：



考虑到基本分词功能单一，因此不采用该方法。而对于面向索引分词，特点是充分考虑了歧义句的因素，比如对“主副食品”分词，能够产生[主副食品/n, 主副食, 副食, 副食品, 食品]五个分词结果，鉴于其分词结果会派生出很多新词，会极大增加倒排索引的规模，因此不在本项目总采用。

NLP分词功能强大，能够支持新词发现功能。但是执行时间相对较长，并且非常消耗内存。而精准分词在易用性、稳定性、准确性、以及分词效率上都取得了一个不错的平衡，也是作者亲自推荐的方法（https://github.com/NLPchina/an

sj\_seg/issues/156）。我们经过粗略测试，在同样的软硬件环境下，NLP方法初始化时间约为100秒，而精准分词方法只需10秒左右来初始化。而在内存占用方面，NLP方法相比于精准分词方法占用了极大部分内存，而且其分词时间也更久。

综合上述考虑，本项目中对于索引构建过程中对文本的分词采用精准分词方法，为了保持一致性，对于搜索语句的分词也采用该方法。

**词典**

Ansj默认分词词典有386211个词项，共6MB左右。即便如此，该词典也只涵盖了首字母从A-G的常用词项，不过在实际体验中对分词结果的影响不大。

停用词词典需要自行添加，本项目添加的停用词有1534个，词典大小共10.2KB。

## 3. 检索策略

🡪裕杰

总体介绍

### 3.1 检索流程

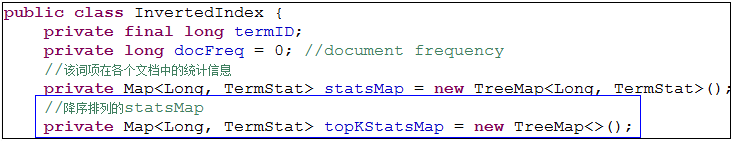
🡪裕杰

流程图

### 3.2 结果排序

**胜者表**

由于倒排索引中已经存储了每篇文档的tf-idf信息，因此，在搜索引起首次启动，加载倒排索引到内存的时候，程序会对每个词项的倒排索引按照和tf-idf值由高到底排序，并且截取出TopK（在configuration.properties里可配置，默认为50）篇文档。只有这TopK篇文档才可以参与最后的相似度运算。



**合并算法**

在对每个搜索词项的相关文档集进行合并之前，程序会先按照相关文档的规模从小到大进行排序，然后再按照该顺序进行相关文档合并，从而减少不必要的合并计算。

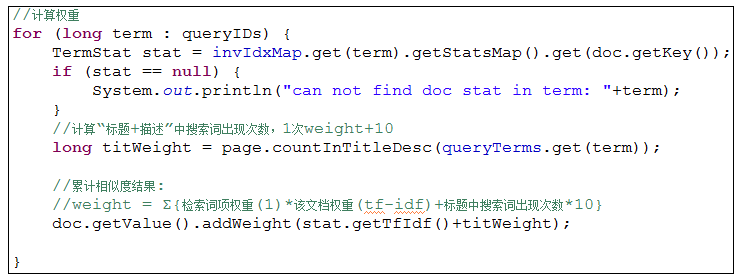


**相似度计算**

本项目权重计算基于tf-idf，并且为了达到更快的速度舍去了log计算。此外，我们考虑到如果搜索词出现在某篇文档的标题或者摘要中，那么该文档的相关度应该更高。因此我们采取的相关度权重算法为：

其中，括号中前加号前半部分是nnn.ntn算法，而后半部分则是考虑到搜索词在“标题+摘要”中的权重。如果搜索词在标题和摘要中出现了次，那么其权重将会增加。而具体是增加100分的权重或者是更多，有待日后不断调试，以获得最优方案。

从前端显示的角度来看，该算法有利于提升用户体验。因为如果某篇文档标题或摘要中出现多次搜索词，那么基于该权重算法，该文档排名会更靠前，并且通过关键词高亮效果，用户能够轻松获取相关的结果。



### 3.3 结果聚类

🡪裕杰

聚类探究与算法实现

## 4. 前端处理

🡪高妍

总体介绍，功能实现

### 4.1 页面元素

🡪高妍

截图解释

### 4.2 自动补齐

🡪高妍

### 4.3 搜索词推荐

🡪高妍

### 4.4 摘要快照及高亮

🡪高妍

# **三．测试与评估**

本项目测试主要分为两个部分，功能测试和性能测试。其中功能测试包括了黑盒、白盒及回归测试，涵盖了项目开发过程中主要模块的功能测试以及bug修复统计。性能测试包括了索引构建、服务启动和检索过程中分词、排序、聚类等每一步的时间开销。

## 1. 测试环境

**硬件环境：**

CPU：Intel Core2 Duo 2.2GHz

Memory：3.0GB DDR2

**软件环境：**

操作系统：Windows 8.1 Enterprise

服务器：Apache Tomcat 7.0.55

编译器：Eclipse 4.3 Kepller

数据库：Mysql 5.1

JDK：1.7

**语料集：**

来源：网易体育，腾讯体育，搜狐体育，MSN Sports

Excel表格

## 2. 测试结果

### 2.1 功能测试

🡪高妍

排序、聚类、自动补齐、相关搜索推荐、snippet、快照等

### 2.2 性能测试

🡪陈新荃

启动速度、响应速度

Excel表格

# **四．创新点**

## 1. 相似度计算

除了在结果排序过程中使用到了胜者表以及合并算法进行优化之外，本项目还对相似度计算进行了创新。

程序中文档的权重计算基于tf-idf，并且考虑到搜索词如果出现在某篇文档的标题或者摘要中，那么该文档的相关度应该更高。因此我们采取的相关度权重算法为：

其中，括号中前加号前半部分是*nnn*.*ntn*算法，而后半部分则是考虑到搜索词在“标题+摘要”中的权重。如果搜索词在“标题+摘要”中出现了次，那么其权重将会增加。目前来看，基于本项目语料集，每出现一次增加100分的权重，其效果甚是不错。

## 2. 中间字自动补全

🡪高妍

## 3. 【还有什么尽管添加】

# **五．经验与总结**

## 1. 个人经验总结

**陈新荃：**

第一次像模像样的带领一个团队完成一个项目。花了很多时间和精力在这个项目上，但是获益良多。经历过这样一个项目，除了信息检索相关的知识得到巩固和加强之外，在编程能力上也有了提高，代码的健壮性和可读性方面得到了加强。

深深感受到了项目开发过程的复杂，除了要理清思路把握项目整体的方向和进展，还需要考虑成员之间的分工合作。大到指定开发计划，小到考虑某段代码的开销等等，都需要清晰地思路和周全的考虑。这些对我来说都是历练。在此非常感谢团队成员的积极配合，项目能够圆满完成是大家齐心协力的结果，能与大家一起共事我深感荣幸。

**高妍：**

**林裕杰：**

**肖卡飞：**

## 2. 组长总结

本项目开始于2014年9月23日(第一次会议)，并于12月14日圆满完成。

我们团队成员来自两个不同学院，在这几个月的期间里，虽然大家都学业繁忙，但是所有人都积极参与，发挥各自所长，使得项目进展有条不紊，并且最后完成了一份不错的作品。非常感谢团队所有人的付出与努力，并希望以后能够再有机会一起共事！