**基于Lucene的分析与应用**

**实现方案修改**

Version1.0

小组成员：

刘宏宇

滕延林

顾泽鹏

杨帆

周晓懿

**版本变更记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 变更时间 | 修改人 | 审核人 | 备注 |
| 1.0 | 20160420 | 刘宏宇 | 刘宏宇 顾泽鹏 杨帆 滕延林  周晓懿 | 落实改进方案的修改 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1. 前言 4](#_Toc448949853)

[1.1目的 4](#_Toc448949854)

[1.2项目计划调整 4](#_Toc448949855)

[2. 基本框架搭建 5](#_Toc448949856)

[2.1Lucene导入 5](#_Toc448949857)

[2.2文本入库 5](#_Toc448949858)

[2.3文本检索 5](#_Toc448949859)

[3. 扩展功能 7](#_Toc448949860)

[3.1中文分词 7](#_Toc448949861)

[3.2图像检索 7](#_Toc448949862)

[4. 性能优化 8](#_Toc448949863)

# 前言

### 1.1目的

在落实项目计划的具体细节时，根据工作进展情况与未来预期，相应调节项目的实现方案。在有限的时间内完成指定目标，并指导后续工作。

### 1.2项目计划调整

调整前功能

|  |  |  |
| --- | --- | --- |
| 功能需求 | 性质 | 备注 |
| 文本入库 | 基本需求 | 文件操作，已实现 |
| 文本检索 | 基本需求 | Lucene自带API，已实现 |
| 智能化检索 | 提高需求 | 对于数据进行解析，得到更加智能化的检索结果 |
| 非结构化数据管理 | 提高需求 | 数据四面体结构 |
| 数据挖掘 | 提高需求 | 文本摘要、分类等 |

调整后功能

|  |  |  |
| --- | --- | --- |
| 功能需求 | 性质 | 备注 |
| 文本入库 | 基本需求 | 文件操作，已实现 |
| 文本检索 | 基本需求 | Lucene自带API，已实现 |
| 中文分词 | 提高需求 | 对分词器进行改进，已实现 |
| 图像检索 | 提高需求 | 提取图像特征，然后进行编码并检索，已实现 |
| 哈希 | 性能优化 | 提高检索速度 |

调整原因

|  |  |  |
| --- | --- | --- |
| 功能修改 | 调整方式 | 调整原因 |
| 智能化检索 | 删除 | 缺少统一标准，概念比较模糊，测试难度大，比较耗时 |
| 中文分词 | 增加 | 在处理中文时，必须用到中文词汇作为索引表表头，而Lucene中没有中文分词功能 |
| 图像检索 | 增加 | 许多文档中含有图像，而这些图像往往包含重要信息。此外图像类型的数据十分丰富，而Lucene并不支持图像的检索 |
| 哈希 | 增加 | 为了提高系统的性能，降低检索时间 |

# 基本框架搭建

### 2.1Lucene导入

Lucene是使用Java语言编写的开源代码。首先搭建Lucene的运行环境，在Windows7上安装JRE6.0，然后将Lucene3.6.2源代码导入eclipse。

由于该项目涉及到文件系统的操作，还要在系统中（D盘）专门分出一块区域作为文件系统。

### 2.2文本入库

用户在页面输入将要入库的文件的路径，文件将复制到指定的文件夹内。完成文本入库。

### 2.3文本检索

用户在页面输入检索关键字，在页面会返回检索结果。即包含这些关键字的文档，点开文档，关键字还会有高亮显示。

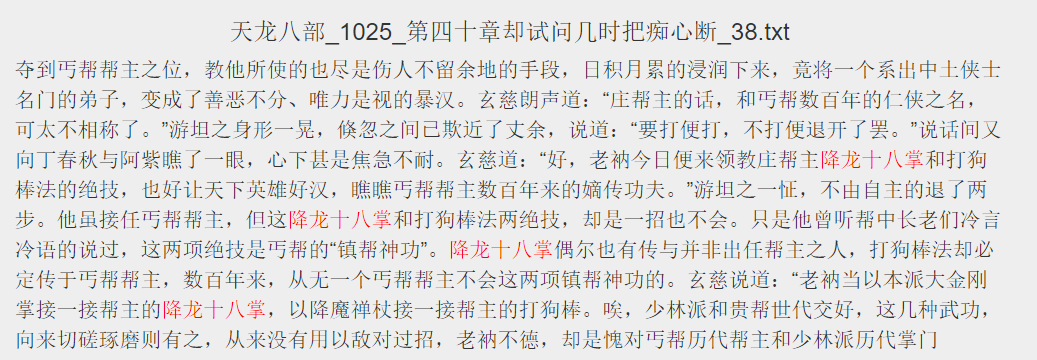
比如搜索“降龙十八掌”



点击查询，出现搜索结果



可以看到，在多部小说中，曾经出现过“降龙十八掌”这个词，并且按照相似度进行排序。点击链接，可以看到具体的信息。



可以看到，文本中的待查询的关键词都已经用红色字体进行标注。

# 扩展功能

### 3.1中文分词

中文分词指的是将一个汉字序列切分成一个一个单独的词。分词就是将连续的字序列按照一定的规范重新组合成词序列的过程。在英文的行文中，单词之间是以空格作为自然分界符的，而中文只是字、句和段能通过明显的分界符来简单划界，唯独词没有一个形式上的分界符，虽然英文也同样存在短语的划分问题，不过在词这一层上，中文比之英文要复杂的多、困难的多。

在进行分词时，首先标点符号是天然的分割符号。然后，根据已有的词库，根据词频来进行分词。首先将一句话从头到尾读入，然后将高频词匹配出来。这样做的好处的简单，缺点是对于低频词匹配较困难。但考虑到低频词本身就很难作为检索的关键字出现，这样做也不会过分减低效果。

### 3.2图像检索

基于内容图像检索是根据图像的底层特征进行检索的，相比于数据库中的检索方式，基于内容图像检索系统有着很大的差异，传统数据库中的检索必须是精确匹配，而基于内容图像检索系统是近似匹配。

该系统在实现图像检索流程有以下三个步骤：

步骤一：特征提取。根据用户需求，选择合适的图像特征。对于数据库中的全体图像，统一进行特征提取，特性以特性向量的形式表示，并存储在数据库中，形成图像的特征库。该步骤在图像入库时进行，降低检索时间。并使用Lucene对于特征码建立索引列表。

步骤二：映射到特征空间。当用户提出检索请求时，将待检索图像表示成特征向量。通过PCA等方法将特征向量降维，并表示成特征空间的点。

步骤三：相似性度量。计算待检索图像的特征向量与数据库图像的特征向量的“距离”，常用欧式距离、余弦距离、L1距离等。根据距离大小，将数据库图像进行升序排序，选取距离最小的一批图像作为检索结果。

# 性能优化

本程序作为检索系统，需要支持对大规模数据的索引及搜索，因此需要具有很高的效率。这主要体现在两个方面：

1. 高效实现大规模数据实时索引入库

作为一个全文检索系统，如果入库数据占据了大量时间，那么必然影响了检索系统的实时性。因此程序需要高效完成大规模数据的索引入库过程。

1. 迅速响应查询条件并返回结果

衡量搜索引擎性能的一个重要指标就是响应时间，因此程序需要能够在很短的时间内响应查询请求并返回检索结果

本系统将采用哈希的方式，对于编码进行处理，以便于快速索引。