第34卷 第16期 计 算 机 工 程 2008年8月
Vol.34 No.16 Computer Engineering August 2008
·软件技术与数据库· 文章编号：1000—3428(2008)16—0092—03 文献标识码：A 中图分类号：TP311.5

基于 Lucene 的语义检索系统

郑 廷，郑 诚
（安徽大学计算机科学与技术学院教育部重点实验室，合肥 230039）

摘 要：在一种基于 LUCENE 的传统文本检索引擎之上，采用 C/S 架构模式的语义检索实验系统。用户可以根据需要，从客户端向服务器提交相应的查询信息配置，服务器根据此配置，通过本体导航与同义词查询 2 种查询扩展优化技术，对提交的查询关键词组进行查询、扩展等优化处理后，将经优化处理过的查询关键词组导入传统的文本检索引擎中，对文档资源进行匹配，将查询结果根据用户要求的排列，并依次返回给用户。通过用户与服务器的信息交互与对查询语句的查询扩展，该系统提高了查准率与查全率。
关键词：文本检索；本体；同义词；查询扩展；C/S 架构；语义

Semantic Retrieval System Based on Lucene

ZHENG Ting, ZHENG Cheng
(Key Lab of Intelligent Computing & Signal Processing , Anhui University, Hefei 230039)

【Abstract】With a traditional text search engine which is based on Lucene, this paper describes a semantic search experimental system which uses C/S pattern. In the system, user sets the configuration which will be passed to the server according to user's wants, and the server operates the query keywords which is submitted by user with ontology navigation and synonym searching according to these configuration, enters these optimized query keywords into traditional text search engine for searching the result, and returns these results to client in the order defined by user. This system uses the interaction between the client and the server, and query expanding trying to raise rate of precision and recall of the retrieval system.
【Key words】text retrieval; ontology; synonym; query expanding; C/S pattern; semantic

1 概述

传统信息检索技术一般是对于关键词的字符匹配和全文检索技术，主要借助目录、索引和关键词等方法来实现。此技术简单、快捷和容易实现，但其存在“忠实表达”问题。由于在大多数情况下，用户很难通过简单的几个关键词来忠实地表达其检索需要，因此检索质量不尽人意。

由于提交的查询语句的歧义性和不明确性，返回给用户的检索结果并非用户需要的资源。针对这种缺陷，该实验系统利用专业领域本体导航[1]与同义词查询扩展技术对传统文本检索引擎的资源检索过程进行优化。获取包含与用户提交的原查询关键词具有语义关系的文本资源，使返回给用户的检索结果更加忠实于用户原本的真实含义的文本资源。

2 实验系统的设计

该实验系统主要由客户端和一个文本检索服务器组成。用户通过客户端主要负责向服务器发送用户提交的查询关键词与相关的参数值，文本检索服务器根据客户端提交的查询语句，对其进行分析后，获取对应的查询关键词，并根据用户提交的优化参数决定是否对其进行优化扩展[1]。

2.1 查询客户端

整个实验系统采用基于C/S的框架结构，因此，客户端主要负责用户与文本检索服务器之间的信息交互，其提供两方面的作用：用户提交相关查询信息到服务器；获取自服务器返回的与用户查询相对应的检索结果列表[2]。查询信息提交页面是用户向检索服务器提交查询信息的平台，查询信息大致可分为两部分：用户查询语句和相关的查询配置。实验系统流程见图 1。其中，用户的查询语句即用户提供到检索服务器以供参照的字、词组或句子。

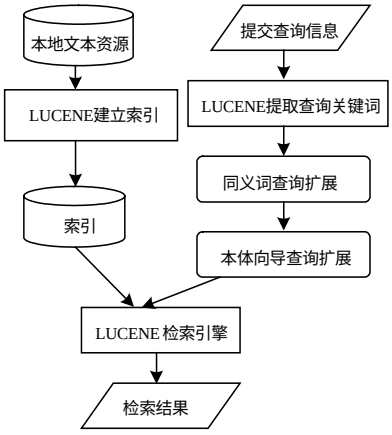

图 1 实验系统流程

2.2 文本检索服务器

文本检索服务器的主要功能是对用户提交查询语句进行分析，其内部结构主要由 LUCENE 传统文本检索系统与查询关键词扩展两大部分组成，其中，查询关键词扩展优化部分又分为 WORDNET 查询扩展与本体导航查询扩展两种。其主要工作流程为：先从查询语句中提取查询关键词，根据用户

基金项目：安徽省自然科学基金资助项目(050420204)；安徽省高校自然科学研究基金资助项目(2006kj055B)
作者简介：郑 廷(1981－)，男，硕士研究生，主研方向：数据挖掘，语义 Web；郑 诚，副教授、博士
收稿日期：2007-09-02 **E-mail：**tizheng361@126.com

需要进行扩展优化，通过传统文本检索引擎、经过优化的查询关键词，对待检索的文本资源进行检索。其中，对于用户提交的查询关键词的优化主要分为：同义词查询扩展和本体导航查询扩展，两者是根据用户的选择而决定是否使用的，用户可在查询信息提交页面的“同义词扩展”与“语义相关度”两个选项中根据需要进行选择，如果两者都被选用，则先对查询关键词进行同义词扩展优化。同义词和本体导航查询扩展见图 2、图 3。

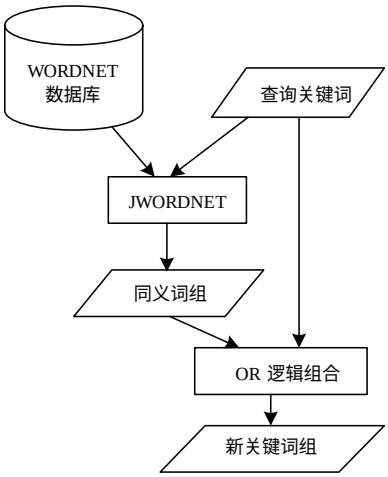

图 2　同义词查询扩展

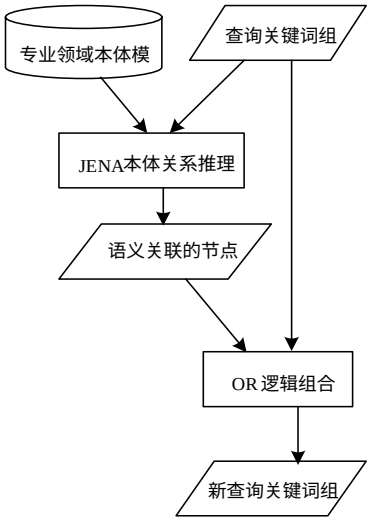

图 3　本体导航查询扩展

3　使用工具及操作

实验系统使用的各工具以及关系如图 4 所示。

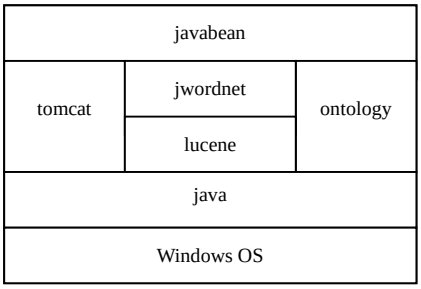

图 4　系统的软件结构

3.1　户查询信息提交

用户查询信息提交的页面由服务器发送到查询客户端的，由用户填写、提交的JSP页面[2]分别由以下部分组成：

(1)同义词扩展

选择后，服务器会利用 WordNet 查询关键词组的同义词，并与原有的查询关键词组一同组成新的查询关键词组，并通过 WORDNET，并进入下一处理环节。此时则完成利用 WORDNET 获取原始关键词同义词的查询扩展。

(2)语义相关度

此参数用于控制本体模型中语义层上的查询扩展，可根据用户选择向服务器提交 5 个参数值(-2, -1, 0, 1, 2)，分别代表对于本体导航的语义推理程度：

1)正值代表抽象的语义关联查询：2 表示在本体模型中检索查询关键词的祖父级节点；1 则表示在本体模型中检索查询关键词的父节点。

2)0 代表不对查询关键词做语义上的查询扩展。

3)负值代表在本体模型中做具体的语义关联查询：-1 表示在本体模型中检索查询关键词的子节点；-2 表示在本体模型中检索查询关键词的孙节点。

根据语义相关度参数，而通过本体导航获得的新的关键词和原有的查询关键词组组合成为新的查询关键词组。

(3)检索结果的排列顺序

此参数决定服务器在根据以上两种参数完成对于资源的检索后，返回给客户端并显示给用户的检索结果排列方式。分别为按照文件最近修改时间排序，按照 LUCENE 的相似度算法排序和文件名称的字母顺序排序。

(4)本体模型文件地址

指定本体向导查询扩展使用的本体模型的文件本地路径。

(5)待检索文本资源地址

指定待被检索文本资源的本地路径。

3.2　查询关键词的分析提取

查询关键词的分析提取，是在用户提交的查询语句的基础上，利用LUCENE开发包的分析器StandardAnalyzer，StandardAnalyzer是LUCENE开发包中内置的一种Analyzer的实现，可以将其理解为“标准分析器”，它包含LUCENE内置的几种分词器和过滤器，可以进行智能识别，如：字母，缩略语，公司名称，电子邮件地址，计算机主机名称，数量，即与内政部符号，序号，IP地址，CJK宏(中文、日文、韩文)字符[3]。

3.3　文本资源索引的建立

利用 LUCENE 对文档数据建立索引，具体步骤如下：

(1)把文档数据转化为 LUCENE 可以处理的类型，本实验系统所要检索的文档数据默认为.txt 纯文本文件，所以，不需要考虑文档类型的转化，可直接建立 Document。

(2)对文档数据进行分析。

(3)调用 IndexWritter 中 addDocument()的方法使 LUCENE 建立索引。

建立Document类型，以便将其文本资源转变为LUCENE可以识别的Document模式，向新建立的Document实例中添加“path”，“title”，“contents”和“modified”4 个field内容，分别代表着被检索文档的路径、标题、文档内容、最后修改时间。为文档资源建立索引，并存储于硬盘中[3]。

3.4　查询关键词的扩展

3.4.1　同义词的获取

同义词的获取是查询关键词组扩展优化选择之一，在从用户提交的查询语句中提取最初的查询关键词组后，根据用

户的选择，以决定是否对最初的查询关键词组从WORDNET的词典中获取相对应的同义词，以便对查询关键词进行扩展优化[4]。此时利用的JWORDNET作为从WORDNET词典中提取同义词的接口，方法wnexpand(String[] queryKeywords)核心代码如下：

```
…
int i=0, j=0;
Synset[] [] senses;
DictionaryDatabase dictionary = new FileBackedDictionary();
While(queryKeywords[j]!=null){
Init();
IndexWord word = dictionary.lookupIndexWord (POS.NOUN, queryKeywords[KEYNUM]);
senses=word.getSense();
addtempsenses(senses); j ++;}
addqueryKeyWords(String tempsenses);
…
```

3.4.2 本体导航

由于继承(inheritance)是本体中最简单的关系，其相关概念即是通过具体化操作获取更抽象概念的子概念，继承关系通常将本体的分类结构定义为一个包含父节点以及与其有isA关系的子节点[5]。通过继承关系，本文使用了“具体化”(focalization)与“抽象化”(generalization)两种语义处理方法进行分类导航：前者定义程序为F(c)={ci | ci=isA(c)}}。其中，*c*表示本体中的一个概念。后者定义程序为G(c)={ci|c=isA(ci)}。

具体步骤如下：

(1)查询关键词的抽象化

作为抽象化优化，即在本体模型文档中将查询关键词添加命名空间的前缀后，视其为本体资源中一组状态(statement)的三元组(triple)的宾语(obejct)，其谓语即属性部分为所属子类(subClassOf)，从而确定其主语(subject)部分的资源。

核心代码如下：

```
…
public static void issubClassof( PrintStream out, OntClass cls ) {
for(Iterator i=cls.listSuperClasses(true);
  i.hasNext(); ) {OntClass c=(OntClass) i.next();
if(!c.isAnon()){PrefixMapping prefixes=c.getModel().getGraph().getPrefixMapping();
String shortform=prefixes.shortForm(c.getURI());
Removens(shortform);
addqueryKeyWords(shortform);}}}
…
```

(2)查询关键词的具体化

与抽象化比较起来，则将原始的关键词作为三元组的主语，关键代码如下：

```
…
public static void issuperbClassof( PrintStream out, OntClass cls ) {
for (Iterator i = cls.listSubClasses( true ); i.hasNext(); ) {OntClass c=(OntClass) i.next();
if(!c.isAnon()){PrefixMapping refixes=c.getModel().getGraph().getPrefixMapping();
String shortform=prefixes.shortForm(c.getURI());
Removens(shortform);
addqueryKeyWords(shortform);}}}
…
```

4 实验结果

本实验系统采用了同义词查询扩展，从而提高了检索结果的查全率，并通过与用户的信息交互，使用户可以根据需要，获取更加贴近其提交查询的“忠实表达”。

实验举例 1：用户希望从本地的文档资源中查询与一种名叫“ROSA”匹萨相关的资源，但是却并不完全清楚这种匹萨的名字拼写，利用系统内提供的本体向导的查询扩展技术，通过对 PIZZA 本体模型的推理检索，而获取用户提交的查询进行语义扩展的优化，即根据语义相关度参数-2，在本体模型进行具体化查询扩展以获取查询关键词“PIZZA”的孙节点，即“NamedPizza”的子节点中所包含“ROSA”的资源，并将其添加到查询关键词组中，由此用户从返回的检索结果中获得与“ROSA”以及其他同类匹萨相关的资料，从而了解到“ROSA”的正确拼写，便可以重新提交“ROSA”的查询语句，进一步获得更加精确的检索结果，最终有效地提高了检索结果的查准率。

实验举例 2：用户提交的查询语句中包含名“queries”，利用本文的实验系统中包含的 JWORDNET 的查询扩展后，利用 Synset()返回的同义词组中，不仅包含有与此关键词同义的单词，还包含有其原形“query”，一同添加到查询关键词组中，则既可获得包含“queries”的文档，也可获得包含“query”和其同义词组的文档资源，从而提高整个检索系统的查全率。

5 结束语

本文实验系统在基于传统文本检索的基础上，利用同义词查询扩展与本体导航查询扩展两种技术，对从用户提交的查询语句中提取的关键词组进行了优化，分别提高了查全率与查准率，并利用 C/S 架构形式，实现用户与服务器之间的信息交互，使用户可以根据检索返回的结果，修改提交的查询信息，以便获得更加贴近所需的文档资源。下一步的工作为：建立文档的预处理格式，实现对于多种文件格式的文本检索；对于待检索的文档资源进行预处理，利用 VSM 为文档中包含的出现频率较高的词汇赋予权值，以便提高检索效率，并尽量返回给用户相关度较高的检索结果，以此控制由于查询扩展优化而造成的检索结果过多的问题。

参考文献

[1] Bonino D, Corno F, Farinetti L. Ontology Driven Semantic Search[J]. WSEAS Transaction on Information Science and Application, 2004, 6(1): 1597-1605.

[2] Koide S, Kawamura M. Semantic Search An Implementation, Deployments and Lessons Learned[C]//Proc. of ASCW’06. Beijing, China: [s. n.], 2006.

[3] Hatcher E O. Lucene in Action[M]. [S. l.]: Manning Publications Co., 2005: 46-183.

[4] Miller G A. WordNet: A Lexical Database for English[J]. Communications of the ACM, 1995, 38(11): 39-70.

[5] McBride B. An Introduction to RDF and the Jena RDF API[Z]. (2007-03-21). http://jena.sourceforge.net/tutorial/RDF_API/index.html.