**学习Xapian(1) – 基础的建索引和搜索**

[2 Replies](https://www.coder4.com/archives/2218#comments)

**前言：**

Xapian是一款开源的C++信息检索系统，提供了非常强大的功能。

国人喜欢跟风，收到某宴的影响，国人一直推崇Sphinx：与MySQL深度集成，开箱即用，非常傻瓜。但是它定制起来非常的麻烦，就连最基础的中文分词都要改好多地方才能实现。

与市面上其他的信息检索相比，Xapian类似于Lucene，提供丰富、可拓展的编程接口，让Xapian能够更好的融入你的系统中。同时，他的检索性能又远远高于Lucene，并采用BM-25模型，具有更好的检索效果。

至于大家最关心的中文问题，我可以负责任的说：虽然Xapian无内置的中文分词算法，但是核心与分词部分毫无耦合，只要借助外置分词器，即可无需改动任何Xapian代码，实现对中文文本的索引和检索。

另外：Xapian的发音为Zap-in，X读Z，大家不要搞错了。（这是常识了，比如XEN发音就是Zen）

本系列文章不会说的很细，目标受众是：  
1)熟悉信息检索的基本知识，如TF、IDF、BM25模型等  
2)使用过至少一种开源信息检索系统，如Lucene

Xapian的教程非常精简，Documentation又是Doxygen生成的，我接触Xapian也刚刚1天，因此很多地方都不是很详细。我也是抱着边写边学的心态，才能写下这一系列的文章，如果有不对的地方，欢迎大家指正！

**1、常用的数据结构**

检索相关：  
Xapian::Database 用于读取索引。  
Xapian::Enquire 提供检索服务，与Xapian::Database配合使用  
Xapian::QueryParser 查询语句解析器  
Xapian::Query 查询语句  
Xapian::MSet 检索返回的匹配结果记录集

建索引相关  
Xapian::WritableDatabase 用于建立索引。  
Xapian::TermGenerator 非常简单的切词、建索引器，不是必须使用的，可用其他替代，但是提供了一些帮助函数，非常好用。

共用：  
Xapian::Document 文档的抽象。  
Xapian::SimpleStopper 停用词  
Xapian::Error 异常类，.get\_description()获取详细信息。

**2、建立索引的步骤**

(1)打开一个Xapian::WritableDatabase  
(2)准备Document  
可以用.set\_data(string)设置负载数据，有且只能有一个  
可以用.add\_value(slot, string)设置附加域，可以有N个，slot不能是-1。  
上述两个都是Document中附加的，只存储，不会被分词和索引。  
建立索引域的方法有两种  
a)Document.add\_term(word, pos)，一个词+位置，一个词+位置的放入。这样显然有点麻烦，于是有了b)方式。  
b)建立Xapian::TermGenerator，.set\_document(doc)，然后将一段**用空格分开的字符串**传入index\_text。之后，doc中就会有这段文本的索引域了。  
注意这里我说的是**用空格分开的字符串**，也就是说，中文分词的步骤我们可以直接拿到外面去做，处理好了再交给Xapian即可！  
(3)Document准备完毕后，将其加入DB  
(4)切记，一定要DB.commit()，这和Lucene是一样的。

**3、检索流程**

与建索引相比，检索要复杂的多。  
(1)打开Xapian::Database，路径是建索引时候的WritableDatabase的路径。  
(2)用Db构造Xapian::Enquire，后者提供了检索的接口。  
(3)使用Xapian::QueryParser解析Query字符串，生成Xapian::Query。  
(4)enquire.set\_query()，进行检索，说实话这个API真够冏的……  
(5)enquire.get\_mset(start,len)返回结果集合，MSet，相当于支持分页，这个太赞了！  
(6)对返回的MSet遍历使用Xapian::MSetIterator，get\_rank()获取排名，get\_document返回检索时对应的文档。

**4、检索语法**

检索语法：  
Term AND|OR|NOT Term …  
Term -> Term|~Term  
其中~表示同义词（依赖于建索引时候额外提供的同义词pair）

**5、关于域的支持**

最早的Xapian是不支持多个域的，现在虽然支持，但是没有提供类似Lucene中add\_field这样的操作，而是采用了“前缀”和“映射”的方法。  
它依赖于建索引和检索时候分别提供额外的“前缀”和“前缀映射”。  
(1)建立索引时，以Xapian::TermGenerator为例  
需要设置TermGenerator.set\_database(db)  
建索引域时，API如下：  
index\_text (const std::string &text, Xapian::termcount wdf\_inc=1, const std::string &prefix=std::string())  
后两个是有默认数值的，第二个是tf增加量，最后一个是前缀，比如Title域，我们令prefix=“T”,Content域，prefix=”C”，这个前缀得是你自己定义好的，并且要前后一致！  
(2)搜索时候，在Xapian::QueryParser中添加映射，.add\_prefix(“title”, “T”)，于是，我们给qp.parse\_query的查询字符串就可以包含域了！形如：

title:新闻 AND content:男篮

包含了两个域！转载自：[学习Xapian(1) – 基础的建索引和搜索](http://www.coder4.com/archives/2218)

基础的部分就说这么多，看代码吧。

源代码：建立索引



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48 | //[原创]四号程序员 http://www.coder4.com  #include <xapian.h>  #include <iostream>    using namespace std;    #define CONTENT "70比 69， 这是 男篮 亚锦赛 历史上 的 最小 分 差 比赛 ， 笑 到 最后 的是 东道主 中国队 。 可以说 ， 这是 一次 最 惊险 的 胜利 ； 也可 以 说 ， 这是 中国男篮 最 幸运 的 结局 。终 >  场 哨 响 ， 中国队 主教练 邓 华德 和 篮管中心 副主任 胡 加时 紧紧拥抱 在一 起 ， 两人 都 激动 得 热泪盈眶 —— 中国队 赢了 ， 赢得 很 庆幸 。"    #define TITLE "这 是 一条 新闻"    #define INDEX\_PATH "./index\_data"    #define F\_DOCID 1    int main()  {      try      {          //Text to be indexed          string content(CONTENT);          string title(TITLE);            //Open an Database for write          Xapian::WritableDatabase db(string(INDEX\_PATH), Xapian::DB\_CREATE\_OR\_OPEN);            //Prepare TermGenerator, just split word by space, not chinese analysis          Xapian::TermGenerator indexer;            //Make Document          Xapian::Document doc;          doc.add\_value(F\_DOCID, string("1104"));          doc.set\_data(content);          indexer.set\_document(doc);          indexer.index\_text(title, 1, "T");          indexer.index\_text(content, 1, "C");            //Add Document to db          db.add\_document(doc);            //Flush to disk          db.commit();      }      catch(const Xapian::Error &e)      {          cout << e.get\_description() << endl;      }      return 0;  } |

转载自：[学习Xapian(1) – 基础的建索引和搜索](http://www.coder4.com/archives/2218)

源代码：检索



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50 | //[原创]四号程序员 http://www.coder4.com  #include <xapian.h>  #include <iostream>    using namespace std;    #define QUERY "title:新闻 AND content:男篮"    #define INDEX\_PATH "./index\_data"    #define F\_DOCID 1    int main()  {      try      {          //Query          string query\_str(QUERY);            //Open an Database for read          Xapian::Database db(string(INDEX\_PATH));            //Open Search Handle          Xapian::Enquire enquire(db);            //Parser Query          Xapian::QueryParser qp;          qp.add\_prefix("title", "T");          qp.add\_prefix("content", "C");          Xapian::Query query = qp.parse\_query(query\_str);          cout << "Query is " << query.get\_description() << endl;            //Find top 10 results          enquire.set\_query(query);          Xapian::MSet result = enquire.get\_mset(0, 10);          cout << result.get\_matches\_estimated() << " results found" << endl;            //Print results          for(Xapian::MSetIterator itr = result.begin(); itr!=result.end(); itr++)          {              Xapian::Document doc = itr.get\_document();              cout << itr.get\_rank() << ": docid " << doc.get\_value(F\_DOCID) << ", data " << doc.get\_data() << endl;          }      }      catch(const Xapian::Error &e)      {          cout << e.get\_description() << endl;      }      return 0;  } |

**您可能也喜欢如下文章:**

1. [使用C++和zlib解压缩.gz文件并读入内存](https://www.coder4.com/archives/1707)
2. [两道面试题……](https://www.coder4.com/archives/2103)
3. [使用Crypto++进行RSA加密](https://www.coder4.com/archives/577)
4. [C++大作业–单件类（Singleton）的实现](https://www.coder4.com/archives/355)
5. [[原创]Cassandra快速入门(0.6.12)](https://www.coder4.com/archives/1401)

This entry was posted in [[IR] Lucene && Sphinx && Xapian](https://www.coder4.com/archives/category/ir), [C && C++](https://www.coder4.com/archives/category/cpp) and tagged [Xapian](https://www.coder4.com/archives/tag/xapian), [信息检索](https://www.coder4.com/archives/tag/%e4%bf%a1%e6%81%af%e6%a3%80%e7%b4%a2), [学习笔记](https://www.coder4.com/archives/tag/%e5%ad%a6%e4%b9%a0%e7%ac%94%e8%ae%b0) on [2011-09-26](https://www.coder4.com/archives/2218).