中文叙词表本体的检索实现及其术语学服务研究*

曾新红　林伟明　明仲

(深圳大学图书馆，深圳，518060)　(深圳大学信息工程学院，深圳，518060)

［摘要］本文首先对中文叙词表本体共建共享系统 OTCSS 的项目背景进行了简单介绍，并阐述了实现中文叙词表本体网络术语学服务（OntoThesaurus-TS）的意义。随后详细描述了 OntoThesaurus 的检索实现方法，以及它的术语学服务应用场景典型案例。最后对 OntoThesaurus 的术语学服务提出了进一步的研究计划。

［关键词］中文叙词表本体 叙词表 本体 术语学服务 检索实现 机辅标引 概念检索 智能检索 Jena SPARQL 查询扩展

［分类号］G254

Implementing Retrieval to OntoThesaurus (Chinese Thesaurus Ontology) and Research on its Terminology Service

Zeng Xinhong　Lin Weiming

(The Library of Shenzhen University, Shenzhen 518060, China)

Ming Zhong

（College of Information Engineering of Shenzhen University, Shenzhen 518060, China）

[Abstract] The context of OTCSS (OntoThesaurus Co-constructing and Sharing System) and the significance of Terminology Service with OntoThesaurus are introduced firstly. Then the methods of implementing retrieval to OntoThesaurus are expounded, and the typical application scenarios of OntoThesaurus-TS are illustrated. Finally the further study plan for OntoThesaurus-TS is presented according to the current international trends of TS.

[Keywords] OntoThesaurus, thesaurus, ontology, Terminology Service, OntoThesaurus-TS, retrieval implementation, computer-aided indexing, concept retrieval, intelligent retrieval, Jena, SPARQL, Query Expansion, QE

“中文叙词表本体共建共享系统(OTCSS)”应用本体技术实现了中文叙词表的形式化表示及其扩展（即 OWL 格式的 OntoThesaurus，中文叙词表本体）以及一致性检查机制，并可在中文叙词表本体的共享应用服务中及时采集主题标引员、领域专家、OPAC 用户及其他用户提出的新词/词间关系和其他修订意见，经统计分析后提供给修订专家参考，结合本体推理技术半自动实现中文叙词表本体的动态完善。

本文主要介绍中文叙词表本体的检索实现方法，并对其术语学服务进行了深入研究，详细描述了 OntoThesaurus 网络术语学服务的应用场景典型案例，最后根据国际最新术语学服务研究动态，提出了进一步的开发和研究计划。

OntoThesaurus 的结构和一致性检查机制、OTCSS 系统的设计与实现方法等相关内容请参考本项目资助发表的其他成果论文。

1 实现中文叙词表本体网络术语学服务（OntoThesaurus-TS）的意义

本系统为 OntoThesaurus 开发的网络术语学服务是交互式的、面向用户的服务，功能包

* 本文系国家社科基金项目“基于本体和知识集成实现中文叙词表的升级、共享和动态完善”（项目编号 05CTQ001）和国家自然科学基金项目“视角理论及其在本体集成中的应用”（项目编号 60673122）研究成果之一。

括：多途径一般检索，高级检索（组配检索），叙词款目详细信息展示，获得所需叙词及其相关信息（如：分类号/入口词/英译名/指定关系相关概念等）。用户利用这些服务，可以发现、获得所需的概念术语及其相关信息，并可应用到任何界面（如编目标引、OPAC 检索、搜索引擎、数据库检索等界面），进行检索、查询扩展（英文扩展，同义词扩展，上/下位词扩展，指定关系扩展等）、查询重构、分类/主题标引、翻译等工作。

与传统的词表服务方式相比，中文叙词表本体网络术语学服务的优势是明显的：用户可以通过 Internet 对叙词表本体进行访问，不受时间、地域的限制；叙词表本体是用户可见的、直观的，可以真正成为连接标引者与检索者的桥梁，从而大大提高标引和检索的效率；网络化的服务方式可使叙词表本体超越图书馆专业范围，实现更广泛的共享应用，为广大网络用户提供方便快捷的术语学指导。

人们在信息检索时总是希望能最快最有效地获得最有价值的信息。对于同一概念，不同的用户可能使用不同的术语来进行检索。传统的关键词匹配方式在检索过程中只进行简单的字符串匹配，并不考虑语义信息，缺乏知识理解能力，因此不可能有很高的查全率和查准率。要想从根本上解决问题，必须实现概念检索（或智能检索）。实现概念检索需要知识组织系统（如叙词表）的帮助。中文叙词表本体以形式化模式明确定义了领域内概念的各种属性，以及概念和概念之间的关系，并以使用者共建的形式实现动态更新完善，因此，与传统叙词表相比，它能够更好地解决语义层次上的问题。通过使用 OntoThesaurus-TS，用户可以很方便地从主题词、入口词、分类号、英译名等多种途径查找到所关心的概念，获得此概念及其相关信息，并可将这些信息应用于各种检索系统和应用系统，从而可以在不改变已有系统（或略微修改其多个检索词的接收方式）的情况下，实现从关键词检索到概念检索的飞跃。

2 *OntoThesaurus* 的检索实现方法

2.1 *OntoThesaurus* 的检索问题

实现中文叙词表本体的网络检索是实现其网络术语学服务的基础。*OntoThesaurus* 的检索实现需要考虑的问题有：

1） 选择合适的本体查询语言实现对 OWL 格式 *OntoThesaurus* 的检索。

2） 除了按主题词（入口词）方式进行检索外，还需要考虑实现从分类号、叙词的英译名等途径进行检索，即实现 *OntoThesaurus* 的多途径检索。

3） 由于用户对需要检索的内容存在不确定因素，因此除了提供精确检索模式，还需要提供前方一致（前方匹配）、任意一致（任意匹配）检索模式。另外还需要提供高级检索模式实现多种途径的组配检索。

4） 需要提供友好的检索界面方便用户使用，检索速度必须能够满足用户的需要。

2.2 面向本体的检索技术

实现 OntoThesaurus 的检索有赖于面向本体的检索语言和检索工具的支持。目前存在的本体查询语言主要有[1]：*ICS-FORTH RQL*，*ILRT SquishQL*，*Intellidimension RDFQL*，*RDFPath*，*VERSA RDF* 查询语言，*TRIPLE*，*TMQL*，*Ontopia Tolog*，*OWL-QL*，*RDQL*，*SPARQL* 等。当前，国际万维网联盟 *W3C* 主要致力于 *SPARQL* 检索语言的研究。另一方面，本体的应用软件开发包（如 HP 公司开发的 *Jena*）也纷纷涌现，很多开发包直接提供检索本体的函数。

经过反复试验，我们选择采用 SPARQL 语言和 HP 公司开发的开源软件开发包 Jena 来实现 OntoThesaurus 的检索。

2.3 *SPARQL* 检索语言

SPARQL 检索语言[2]是 *RDQL* 检索语言的扩展，目前是 *W3C* 的工作草案，而将来可能成为 *W3C* 的推荐标准，主要用于查询任何用 *RDF* 表示的资源。它属于 *SPARQL* 类语言，提供了 *SQL-Like* 的操作来方便用户使用[3]。目前 *Jena* 框架已经支持 *SPARQL* 的使用。

SPARQL 检索语言主要通过图形模式的匹配来实现查询功能。最简单的图形模式是三元组模式，并允许变量出现在三元组的任意一个位置上。图形模式还分为必备模式和可选模式。*SPARQL* 可以检索通过逻辑“与”和逻辑“或”运算组成的复杂图形模式。此外，*SPARQL* 还支持扩展后的值测试检索以及约束检索。*SPARQL* 检索语言的结果可以是结果集，也可以是 *RDF* 图。

运用 *SPARQL* 检索语言可以实现关系数据库中最常用的关系操作，包括选择（*SELECTION*）、投影（*PROJECTION*）、并（*SET UNION*）、差（*SET DIFFERENCE*）、笛卡尔乘积（*CARTESIAN PRODUCT*）、连接（*JOIN*）等[4]。*SPARQL* 有 4 种显示结果的方式：*SELECT*、*CONSTRUCT*、*DESCRIBE* 和 *ASK*。*SELECT* 以表格形式返回结果；*CONSTRUCT* 返回一个 *RDF* 图，该图是通过取代一系列的三元组模式中的变量构造的；*DESCRIBE* 返回的也是一个 *RDF* 图，不同的是该图描述了所找到的资源；*ASK* 返回一个布尔值，用来表示是否找到符合条件的结果。

SPARQL 检索语言通过定义 *WHERE* 子句来声明匹配的条件。在条件中可运用 *FILTER* 来对变量作一定的限制，从而达到筛选检索结果的目的。在 *FILTER* 中可与正则表达式结合使用，以此来构造复杂的限制条件。

目前，*W3C* 只致力于研究 *SPARQL* 检索语言。经过试验，我们发现 *SPARQL* 可以用于检索 *OWL* 本体，且检索速度较快，能够满足 *OntoThesaurus* 的检索应用要求。

2.4 *OntoThesaurus* 网络检索的实现思路

我们把 OntoThesaurus 的检索分为一般检索和高级检索。一般检索即通过单个检索途径进行的检索，为了满足用户的不同使用需求，我们将提供多个检索途径（如主题词（入口词）、各种分类号、叙词的英译名等）。高级检索即多个检索途径的组配检索。从实现角度来看，一般检索和高级检索在技术上并无本质区别。而精确检索、前方一致检索和任意一致检索方式则需要有不同的解决方案。

下面主要介绍通过“主题词（入口词）”途径进行检索的解决思路。其他途径的检索实现可参照此解决思路进行。

• 精确检索

首先需要一套 *API* 来支持本文选用的 *SPARQL* 检索语言，而 *HP* 公司开发的 *Jena* 开发包[5]提供了对该语言的支持。（详见 2.5 节）

接下来为此检索途径的精确检索构造 *SPARQL* 语句，代码示例如下：

```
.....
//生成 sparql 查询语句
String queryString = "";
queryString = queryString + "PREFIX rdf:<" + rdf + ">";
queryString = queryString + "PREFIX xsd:<" + xsd + ">";
queryString = queryString + "PREFIX rdfs:<" + rdfs + ">";
queryString = queryString + "PREFIX owl:<" + owl + ">";
queryString = queryString + "PREFIX :<" + defaultNs + ">";
```

```
queryString = queryString + "SELECT DISTINCT ?x ";
queryString = queryString + "WHERE {{?x rdf:type :Concept.
                    OPTIONAL{?x :PinYin ?z.}. FILTER(?x = <" + defaultNs + term + ">)}";
queryString = queryString + "UNION {?x rdf:type :Concept. ?x :HasNTerm ?y.
                    OPTIONAL{?x :PinYin ?z.}. FILTER(?y = <" + defaultNs + term + ">)}";
queryString = queryString + "UNION {?x rdf:type ?s. ?s rdfs:subClassOf :Concept.
                    OPTIONAL{?x :PinYin ?z.}. FILTER(?x = <" + defaultNs + term + ">)}";
queryString = queryString + "UNION {?x rdf:type ?s. ?s rdfs:subClassOf :Concept. ?x :HasNTerm ?y.
                    OPTIONAL{?x :PinYin ?z.}. FILTER(?y = <" + defaultNs + term + ">)}}";
.....
```

在这里使用了 *UNION* 来完成对子类实例的检索。也可以配合 *Jena* 的自定义规则推出子类实例与父类的实例关系，然后再对其使用 *SPARQL* 检索语言进行检索（但此方法需要考虑推理部分的时间复杂度）。

• 前方一致/任意一致检索

前方一致/任意一致检索的实现稍微复杂一些。前方一致是指数据的前方位置部分与检索词一致；任意一致是指检索词与数据的任何位置相匹配。前方一致可视为任意一致的一个特例。SPARQL 提供了与正则表达式相结合来限制检索结果的功能。正则表达式是一个公式，它用某种特定模式去匹配一类字符串。如表达式^，其作用是与字符串开始的地方进行匹配；表达式$，其作用是与字符串结束的地方进行匹配。

基于此功能，可以为前方一致和任意一致检索构造 *SPARQL* 语句。任意一致检索的 SPARQL 语句示例如下：

```
.....
//生成 sparql 查询语句
String queryString = "";
queryString = queryString + "PREFIX rdf:<" + rdf + ">";
queryString = queryString + "PREFIX xsd:<" + xsd + ">";
queryString = queryString + "PREFIX rdfs:<" + rdfs + ">";
queryString = queryString + "PREFIX owl:<" + owl + ">";
queryString = queryString + "PREFIX :<" + defaultNs + ">";
queryString = queryString + "SELECT DISTINCT ?x ";
queryString = queryString + "WHERE {{?x rdf:type :Concept. OPTIONAL{?x :PinYin ?z.}.
                        FILTER regex(str(?x), \"^" + defaultNs +" (.)*" + term +"\") }";
……
```

与精确检索 SPARQL 语句的不同之处是 *FILTER* 的用法。在这里 *FILTER* 与正则表达式结合使用。由于主题词在 *OntoThesaurus* 中是以资源的方式出现的，即主题词前有默认的命名空间，因此在建立正则表达式的时候需要考虑该命名空间。设 *OntoThesaurus* 的默认命名空间是 *http://www.ontothesaurus.com/ontothesaurus.owl#*，前方一致匹配主题词 *term* 的正则表达式可写为：

^ *http://www.ontothesaurus.com/ontothesaurus.owl#term*

任意一致匹配主题词 term 的正则表达式可写为：

^ *http://www.ontothesaurus.com/ontothesaurus.owl#(.)*term*

其中^ 是指匹配串的起始位置；(.)*是指该位置可出现任意多个任意字符。

2.5 *OntoThesaurus* 的检索实现方法

根据前面提出的实现思路，我们运用 *Jena* 开发包结合 *SPARQL* 语言实现了 *OntoThesaurus* 的检索。具体步骤如下：

1） 运用 *Jena* 提供的 *ModelFactory* 来创建一个 *Ontology Model*，调用该 *Model* 的 *read* 方法将 *OWL* 格式的 *OntoThesaurus* 读入 *Ontology Model* 中。

2） 根据具体的检索要求构造相应的 *SPARQL* 查询语句。

3） 使用 *Jena* 中 *QueryFactory* 的 *create* 方法来为具体的 *SPARQL* 查询语句创建查询 *Query*。

4） 利用上一步的 *Query*，使用 *Jena* 中 *QueryExecutionFactory* 的 *create* 方法生成查询执行语句。

5） 执行查询语句，返回结果集。

6） 从查询结果集中获取查询结果并将结果存储于 *ArrayList* 中，返回给调用者。

以下为部分检索代码。

```
protected Collection searchByTerm(String term,int searchType){
    term = term.trim();
    ArrayList result = new ArrayList();

    //生成 sparql 查询语句
    String querystring = “”;
    ……
    ……
    //生成查询
    Query query = QueryFactory.create(queryString);
    //生成查询执行语句
    QueryExecution qexec = QueryExecutionFactory.create(query,model);
    //执行查询语句
    ResultSet rs = qexec.execSelect();
     while (rs.hasNext())
    {
          //获取查询结果
          ……
          ……
    }
    return result;
}
```

2.6 *OntoThesaurus* 检索效果测试

从 *OntoThesaurus* 中分 12 次抽取片段，每个片段所含的三元组个数比前一个片段大约多 2200 个三元组。对每个片段分别进行精确检索、前方一致检索和任意一致检索。每类检索 10 次，获取其检索时间（不包括页面请求和页面响应时间），求出平均值作为该类检索对应某个片段的检索时间，如图 1。

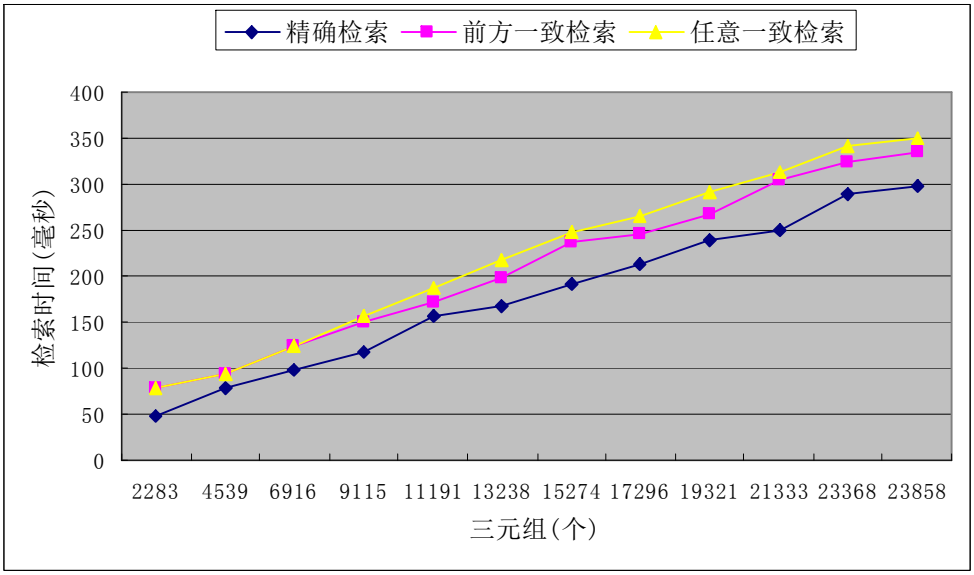

图 1　*OntoThesaurus* 中的三元组数量与检索时间的关系

由图 1 可见，*OntoThesaurus* 中的三元组数量与检索时间大致成正比；任意一致检索与前方一致检索所需时间相差不大，但比精确检索的检索时间要稍长一点；而且在目前两万多三元组的情况下，三种检索的检索时间均在 0.5 秒以内；根据曲线可预测，在三元组数量到达十万级别时，三种检索的检索时间均应在一至两秒内。因此本文对 *OntoThesaurus* 检索问题的解决方案是确实可行的，可满足目前国内大多数现有中文叙词表转换后本体以及新建中文叙词表本体的检索要求。

大容量本体知识库的检索问题是本体相关领域的共同课题，我们将密切关注国际上面向本体的检索语言及支持工具的最新进展，同时也会积极研究各种可能的解决办法（如采用分布式的检索方法）来提高超大型中文叙词表本体的检索速度。

3　*OntoThesaurus-TS* 及其应用场景典型案例

OntoThesaurus-TS 为用户提供基于 Internet 的网络术语学服务界面。用户需输入用户名和密码登录，无用户名及密码者须预先申请账号（一般访客也可使用 Guest 用户登录）。不同用户的检索及术语学服务权限是相同的，仅在提交修订意见的权限上有所区别。

用户可根据自己的具体需要选择一般检索或高级检索（组配检索）模式，可从主题词（入口词）、分类号、叙词英译名等途径进行精确、前方一致或任意一致查询。查询结果为命中叙词列表。单击某一叙词可显示该叙词款目信息，可选择通俗显示方式（适合一般用户）或专业显示方式（适合图书馆专业人士）来显示叙词款目信息。叙词款目中的分类号、相关概念等被设为超链接，可单击这些超链接进行继续检索，如单击分类号可查询该分类号对应的所有叙词，二次检索结果将在右上方窗口中显示。拖动窗口之间的分栏线可任意扩大或缩小分窗口，或单击每个分窗口右上方的小图标激活或取消全屏显示方式，以方便用户浏览和使用。用户如果需要获取叙词、分类号、同义词、英译名或上下位关系词等信息，可在设定获取内容和不同获取内容之间的间隔方式后，单击相应叙词前面的红色购物篮图标获取所需信息，再粘贴至任意的编目著录文本框、检索词输入框或其他文本编辑栏内，进行标引、检索、翻译等活动。

下面介绍几种典型的应用场景。

3.1 帮助标引员更高效地进行主题和分类标引

标引员的任务是对文献进行内容分析，并选择最专指的叙词和分类号对文献进行主题和分类标引。与传统的书本式词表和非广域网电子词表相比，OntoThesaurus-TS 可以为标引员提供更方便快捷的网络词表检索和信息获取服务，从而有效地提高标引员的工作效率和标引质量。标引员可从主题词、入口词、分类号、英译名等途径入手，进行精确一致、前方一致或任意一致（可起到类似轮排索引的作用）查找，并可利用叙词详细款目中的信息进行任意方位的扩展检索，从而可以快速定位到所需的叙词上，获得该叙词及其相应的分类号，同时完成主题标引和分类标引。例如，如图 2 和图 3 所示，利用 OntoThesaurus-TS 从入口词入手查找到正式主题词（叙词），获取叙词后应用于图书的主题标引著录：

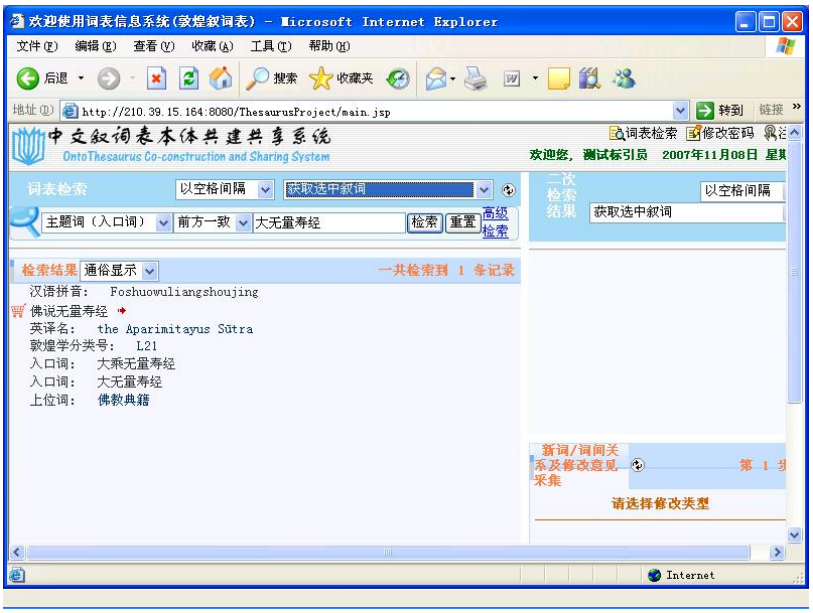

图 2 查找并获取叙词

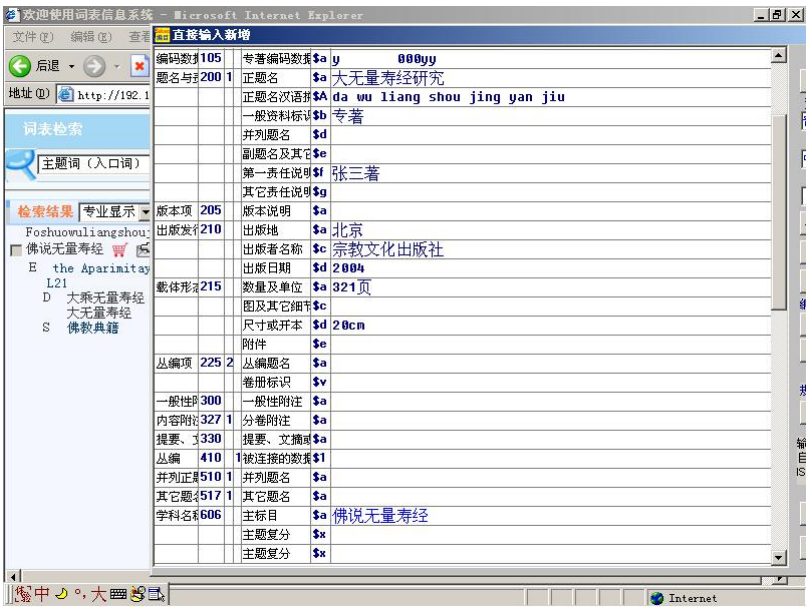

图 3 将获取到的叙词应用于编目中的主题标引著录

3.2 帮助 OPAC 用户进行基于主题或分类的内容检索

图书馆自动化系统通过 OPAC(Online Public Access Catalogue)为用户提供在线目录查询，我们把 OPAC 系统的使用者称为 OPAC 用户。主题检索和分类检索是 OPAC 系统普遍提供的两种重要的内容检索途径，但因为用户对图书馆目录系统所使用的分类法和主题词表（叙词表）不了解，且目前国内的 OPAC 系统一般不提供分类法和主题词表的在线查找服务，所以这两种检索途径在实际使用中应用较少，收效甚微，与图书文献机构在分类主题标引中所投入的巨大脑力劳动极不相称。OntoThesaurus-TS 可向 OPAC 用户在线展示目录系统所使用的叙词表本体，使他们可以非常方便地查找和获取到目录系统所使用的规范叙词和分类号，从而在标引者和检索者之间架起桥梁，减少盲目性，极大地提高这两种途径的实际应用效率。

3.3 一般检索用户利用 OntoThesaurus-TS 获取检索线索

一般检索用户在检索其他数据库或使用搜索引擎查找信息时，可以利用 OntoThesaurus-TS 来启发、获取更多的检索词形态（如规范叙词及其同义词、英译名、关系概念等），从而提高检索的查全率或查准率。例如，如图 4 所示，沿用上例，获取选中叙词及其入口词之后，应用于百度进行搜索：

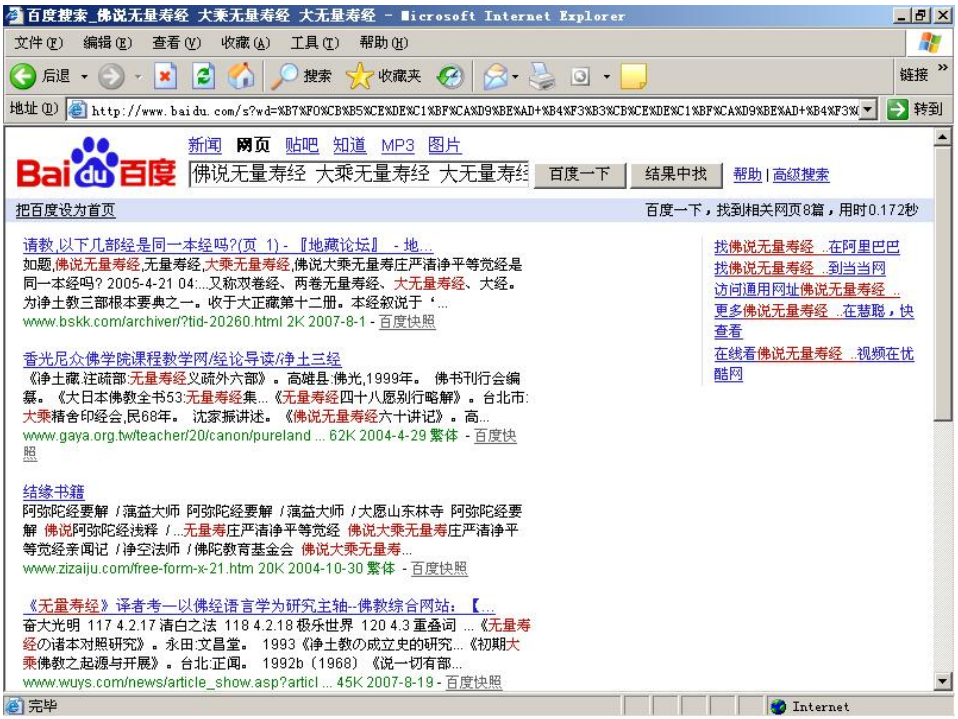

图 4 将获取到的信息应用于搜索引擎

用户可以根据应用系统的要求选择词间间隔方式，目前已实现比较通用的空格或逗号间隔方式，获取到的是无格式文本字符串。我们还会根据不同应用系统的具体要求增加间隔方式（如“OR”或符合某些检索输入规范的方式），获取信息的格式（如 openURL 等格式），以及获取信息的范围（如获取与某叙词对应的指定关系、指定步长的相关叙词等）。

4 进一步的术语学服务研究

在 NKOS 网站近期发布的 JISC state-of-the-art review "Terminology Services and Technologies"[6]中，对术语学服务涉及到的各个方面进行了全面深入的回顾和研究，包括词表（vocabulary）的各种类型、与 TS 相关的各类研究活动和国际计划/项目，以及相关的标准建设，并对该领域所需的进一步活动提出了建议。笔者认为，该文对我国中文术语学服务的研究和建设具有很高的参考价值。

上面介绍的本系统已实现的 OntoThesaurus-TS 功能主要用于满足最终用户（人）的需要。可以通过术语学服务注册（registry）的方式集中提供一系列不同用途的中文叙词表本体供用户自由选择利用。

为了进一步提供面向应用系统的术语学服务，我们在研究了国际上术语学服务的发展趋势的基础上，拟为 OntoThesaurus 开发一套 Web Service API（OntoThesaurus-API），以满足更广泛的 M2M 形式的术语学服务信息交互要求，例如：支持图书馆自动化系统、数字图书馆系统实现机辅标引、自动标引、智能检索、自动映射、语义推理等。

5 结语

目前我们以《敦煌学检索词表》、《社会科学检索词表》（局部）、《中国分类主题词表》（一版局部）等为例实现了中文叙词表本体共建共享系统的原型系统 OTCSS，主要功能已接近实用。实践表明，OTCSS 可用于快速实现已有中文叙词表的本体化升级和共建共享，也可用于从零开始共建共享一个知识组织系统。我们希望今后能与国内的中文叙词表编纂机构或个人合作，共同努力，让传统的中文叙词表不仅能更好地为图书情报界服务，还能在网络信息环境中发挥出更大的作用，成为语义 Web 上中文本体和中文术语学服务的主力。

参考文献

[1] Deliverable 1.3: A survey on ontology tools[EB/OL]. http://babage.dia.fi.upm.es/ontoweb/wpl/OntoRoadMap/documents/D13v1.0.pdf,2005 (Accessed Aug. 17, 2006)

[2] SPARQL Query Language for RDF[EB/OL]. http://www.w3.org/TR/rdf-sparql-query/(Accessed Jul. 23, 2006)

[3] 田庆立，李爱民，方宗德. 应用 RDF 本体图扩充 SPARQL 查询[J]. 情报杂志. 2006，(1)：126-128

[4] Cristian Perez de Laborda，Stefan Conrad. Bringing Relational Data into the Semantic Web using SPARQL and Relational. OWL. Proceedings of the 22nd International Conference on Data Engineering Workshops[C]. USA：IEEE Computer Society Press，2006

[5] Jena-A Semantic Web Framework for Java[EB/OL]. http://jena.sourceforge.net/(Accessed Jul. 7, 2006)

[6] Douglas Tudhope, Traugott Koch, Rachel Heery . Terminology Services and Technology: JISC state of the art review. 15-09-2006. http://nkos.slis.kent.edu/ (Accessed Oct. 17, 2006)