第 33 卷 第 2 期 计 算 机 工 程 2007 年 1 月
Vol.33 No.2 Computer Engineering January 2007
·软件技术与数据库· 文章编号：1000—3428(2007)02—0059—03 文献标识码：A 中图分类号：TP393

针对 XML 文档集的关键词检索结果排序

江腾蛟，万常选
(1. 江西财经大学信息管理学院，南昌 330013；2. 江西财经大学数据与知识工程江西省高校重点实验室，南昌 330013)

摘 要：探讨了针对 XML 文档集中只与内容相关的关键词检索结果的排序问题，针对 XML 文档特征提出了一种新的排序模型，它不同于面向 Web 的 XML 网页的搜索结果的排序。设计了满足这种排序模型的倒排列表索引结构和搜索引擎的体系结构。
关键词：XML；关键词检索；结果排序

Result Ranking of Keyword Search over XML Documents

JIANG Tengjiao, WAN Changxuan
(1. School of Information Management, Jiangxi University of Finance and Economics, Nanchang 330013;
2. Jiangxi Key Laboratory of Data and Knowledge Engineering, Jiangxi University of Finance and Economics, Nanchang 330013)

【Abstract】The paper discusses the problem of efficiently ranking results for keyword search queries related to content-only over XML documents. Evaluating keyword search over XML documents, as opposed to Web-based XML pages, it introduces many new features. This paper presents a new ranking model that is designed to handle these new features of XML keyword search, and designs inverted list indexing structure and search engine's architecture to satisfy the ranking model.
【Key words】XML; Keyword search; Result ranking

可扩展标记语言XML具有自描述性和可扩展性等特点，它已成为最受欢迎的信息表达和数据交换的格式和标准[1]。因此，研究检索XML文档的搜索引擎甚为迫切[2]。搜索引擎技术中尤为重要的是关键词检索技术，这是因为越来越多的普通用户使用搜索引擎，这些用户并不了解数据库，也不了解XML模式（schema），使用最多的是关键词检索。传统的信息检索是以文档或网页为基本单位进行排序的。XML文档中的关键词检索是以XML元素为粒度来返回检索结果的，即在返回检索结果时，并不需要将整个文档返回给用户，而只需返回用户感兴趣且符合检索条件的元素集，该集合可以看作是原文档的一个片段。因此，XML文档中的关键词检索不但可以使得检索结果更为准确，也使得传输的数据量大大减小[1]。

目前，有很多学者对XML检索结果的排序问题进行了研究，但他们大多数是以网页或文档为单位进行排序，没有突出XML查询结果可以为元素粒度的特征。文献[3]研究了以内容为中心（CO）的XML关键词检索结果的排序问题，突出了XML特征，并以网上冲浪模型[7]为基础对检索结果片断进行排序，主要考虑的是用户网上随机点击一个网页以及通过网上的超级链接访问到一个网页的概率，并以此来决定一个网页的重要性。而这一点不太吻合传统信息检索主要以相关性（relevancy）作为排序的重要依据的原则。在文献[4]中，主要讨论的是内容和结构（CAS）的查询，研究了采用加权的词频（tf_w）和倒排元素频率（ief）来计算权重，但它忽略了超级链接和文档内引用等的评判。文献[5]先进行以文档为中心的传统检索，并在此基础上再进行以元素结点为答案结点的结果排序，这样势必影响到检索的性能。

1 XML 文档关键词检索结果的排序模型

首先，给出一个包含会议论文信息的 XML 文档实例，如图 1 所示。

```
01. <workshop date=”28 July 2004”
02.     <title> XML and IR: A SIGIR 2004 Workshop </title>
03.     <papers>
04.       <paper id =”1”>
05.         <title> Indexing and ranking for XML information retrieval </title>
06.         <author> Webb </authou>
07.         <abstract> Indexing and ranking are two key factors for… </abstract>
08.         <body>
09.           <section name=”Introduction”>
10.               As the WWW is becoming a major means of disseminating and …
11.             <subsec name=”XML data model”>
12.                 We model an XML document as an ordered, labeled tree where …
13.             </subsec>
14.               …
15.           </section>
16.           <cite ref=”2”>Querying XML in Xyleme </cite>
17.           <cite xlink=”../paper/xmlql/”>A Query…</cite>
18.         </body>
19.       </paper>
20.       <paper id=”2”>
21.         <title>Querying XML in Xyleme</title>
22.           …
23.       </paper>
24.     </papers>
25.   </workshop>
```

图 1 一个 XML 文档实例

一个 XML 文档可以被建模为一棵有序的标记树，每个元素或属性表示为一个结点，元素-子元素或元素-属性之间的关系用相应结点间的边来表示，如图 2 所示的就是图 1 中

基金项目：江西省教育厅科技基金资助项目(赣教技字[2005]111)
作者简介：江腾蛟(1976－)，女，硕士生、讲师，主研方向：XML 数据管理，信息检索；万常选，博士、教授
收稿日期：2006-04-30 **E-mail**：tj_jiang@163.com

XML 文档片断的 XML 文档树。

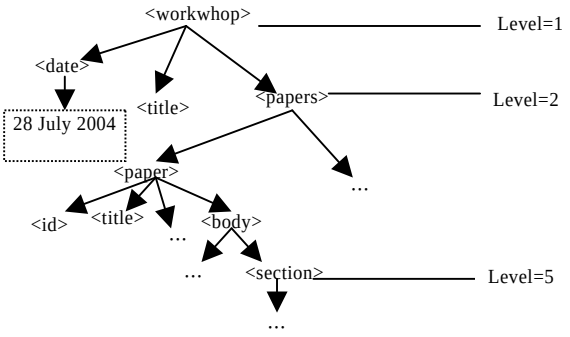

图 2 一个 XML 文档树实例

其次，给出 XML 文档检索的 3 个假设前提（由于篇幅所限，涉及 XML 文档检索的相关技术不能在此详细介绍）。

(1)预设答案结点（Answer Nodes）。文档是具有一定逻辑结构的，这些逻辑结构是由文档作者在设计文档时规定的。因此，不能把结构化文档简单地看作一串字符流，检索信息时不能随意抽取文本片断[8]，而应该返回最符合检索条件、逻辑意义最完整的片断。因此本文的假设前提是文档作者在设计文档时就已给出检索该文档的答案结点，或专家根据领域知识和模式信息已预设了文档的答案结点。

(2)预设位置权重。同一个关键词出现在不同的元素标记（tag）对中，结点语义对结果相关度的影响应该有所不同，如查找关键词“XML”，显然当“XML”出现在标记<keywords>中比出现在标记<paragraph>中的重要性要大得多。这里的假设前提是文档作者在设计文档时就已给出该文档的答案结点的位置权重，或专家根据领域知识和模式信息给出了文档答案结点的位置权重。

(3)如果一个词汇出现在文档的一个片断中，那么这个词汇一定出现在文档的索引表中。这就保证了所有出现在文档中的词汇都能被检索到。

1.1 XML 文档关键词检索结果排序的特征分析

针对 XML 文档集特征，在关键词检索结果排序时主要考虑了以下影响因素：

(1)检索结果的具体性。充分合理利用隐藏在 XML 文档内部的自然的结构信息，检索语义完整的数据片断。检索的结果可以是一个段落、一个章节或者其它粒度级别的元素内容，总之对用户来说，检索结果既要是最具体、最相关的内容，又具有相对独立的逻辑意义，即返回以答案结点为基本单位。

(2)文档的链接和引用。具有广泛被链接的文档（标记 Xlink），它的重要性相对较高，因此它所包含的元素的重要性也相应要高；XML 文档内还有内部引用（标记 IDREF），被引用的元素越多，它的权重也应该更大，同样它们所包含的子元素和它的祖先元素也应有更高的权重。

(3)关键词所在位置。同一个关键词出现在不同的元素标记对中，它所对应的权重应该有所不同。因此对每一个答案结点元素应考虑位置权重。

(4)答案结点的文本大小。当返回文本较大的答案结点结果肯定要逊色于那些提供的答案很简洁的结果。因此，在相关度计算中，还要考虑答案结点文本大小。

(5)关键词之间的距离。对于多关键词检索，还必须考虑关键词之间的距离。这里的距离不仅仅是传统信息检索中提到关键词之间的位移量，还包括关键词所在结点之间的位置，即表现在 XML 文档树中结点间的高度和宽度。

1.2 单关键词检索排序模型设计

首先根据超级链接来计算整篇文档的权重d_w。$d_w = \sum h_i + 1$，h_i表示文档d的一个链入，即d_w的值为链入文档d的超级链接数之和加 1。若一篇文档的权重较大，则相应其内部的所有元素也比较重要。当一篇文档没有任何链入时，则其文档权重为 1。因为本文讨论的是特定的、已存在的XML文档集，所以可排除那些故意增加链入来提升自己文档重要性的不良行为。

接下来考虑文档内的引用。在一个文档内，当一个元素引用另一个元素时，说明被引用的元素比较重要。为了叙述的便利，先给出下面两个假设：(1)设$(f, v) \in FE$，当且仅当元素 f 是对元素 v 的一个引用。(2)设$(u, v) \in CE$，当且仅当元素 u 包含元素 v，即元素 v 是元素 u 的孩子或内容。

设$(f, v) \in FE$，则元素v比较重要；对元素v的引用越多，表明元素v越重要。设$(u, v) \in CE$，$(v, w) \in CE$，如果元素v比较重要，则可推导出元素u和元素w也相应地比较重要。显然，离元素v越远，在XML文档树中表现为层次距离（dist）越大，则其重要性相对要减弱，因此这里设一阻尼系数d_1（$0<d_1<1$）。设$(u, v) \in CE$，u、v之间的层次距离(level值的差的绝对值，见图 2)用$l_{u\text{-}v}$表示，v为答案结点，记r_w为影响答案结点的引用权重，则 $r_w = \sum r_i * d_1^{l_{r_i - v}}$，其中$r_i$表示影响答案结点的一个元素的引用权重，$r_i = \sum ref$ 表示为影响答案结点的某一个元素的引用之和，r_w为所有影响该答案结点的元素的引用权重之和。

接下来考虑另一个影响因素，返回答案结点的文本大小。显然，文本片断越小，搜索者找到所需答案的时间也越少，相应地也希望其排序越前，因此答案结点的权重与结点的文本大小 anstext( )成一定的反比关系。

下面介绍本文中采用的计算答案结点权重的向量空间模型。向量空间模型(Vector Space Model)是目前信息检索最常用的数学模型[2]。在向量空间模型中，将文档和检索的关键词都表示为向量，利用向量的操作，实现文档和关键词的相关度计算。在向量空间模型中，最常用的权重计算方法是$tf \cdot idf$，其中，tf为局部权重，体现了词汇对文档的重要程度，一般用词汇i出现在文档j中的频率tf_{ij}作为tf的值；idf为全局权重，体现了一个词汇区分文档的能力，一般来说，idf的计算公式为：$log(N+0.5)/n$，其中，N表示含有词汇i的文档的数目，n表示数据库中的所有文档的数目。所以，词汇i在文档j中的权重(weight)可以描述为：$weight= tf_{ij}*log(N+0.5)/n$。

根据我们所讨论的XML文档关键词检索的特征，将向量空间模型调整为tf_w*ief，即加权的词频与ief（表示以元素为粒度返回）的乘积。记答案结点的位置权重为tf_d，则整个tf_w必须是综合考虑了文档权重、引用权重、位置权重及词频，$tf_w = d_w*(r_w + tf_d)*tf$，$tf$表示答案结点中的关键词词频。相应地，$ief$的公式调整为：$ief= log(M+0.5)/N$，其中$M$表示文档集内预设为答案结点（AN）的总数量，$N$表示满足该关键词查询的答案结点数量。再结合考虑返回答案结点的文本大小，可将某一个关键词k查询的评分公式完整的描述为

$$tf_k = tf_w * ief / anstext()$$

1.3 多关键词检索排序模型设计

当检索多个关键词$k_1,k_2,\ldots,k_n$时，必须考虑关键词之间的距离问题。显然，关键词越接近，则相关性越高，越符合查询要求。$tf(k_1 \cdots k_n) = \sum_{i=1}^{n} tf_{k_i} / dist(k_1 \cdots k_n)$，其中$tf(k_1 \cdots k_n)$表示多个关键词$k_1,k_2,\ldots,k_n$查询时的总权重，$dist(k_1 \cdots k_n)$表示关键词$k_1,k_2,\ldots,k_n$之间的距离。当$k_1,k_2,\ldots,k_n$出现在同一答案结点内时，$dist(k_1 \cdots k_n)=|k_1,k_2,\cdots, k_n|$；当$k_1,k_2,\ldots,k_n$出现在不同的答案结点内时，容易知道跨过的答案结点越多，语义的逻辑独立性越差，因此我们设一阻尼系数d_2（$0<d_2<1$），$dist(k_1 \cdots k_n) = d_2{}^{l} * | k_1 \cdots k_n |$，$l=l_{low}-l_{an}$，其中$l_{low}$为$k_1$，$k_2,\ldots,k_n$出现在文档树中最下面(即对应的level值最大，见图2)的答案结点的level值，l_{an}为查询结果答案结点的level值。

2 倒排列表的索引结构

利用索引技术检索数据已经被广泛研究，如值索引、结点名索引、边或路径索引、Fabric索引、相对区间坐标索引等；XML数据的编码方案也有位向量编码、前缀编码、区间编码、二叉树编码等[8]。

传统的词条索引（term index）可形式化地描述为：*<term, document-ID, frequency>*。为了适合本文的排序模型，可结合区间编码技术将传统的词条索引修改为扩展的词条索引(eXtend term index,XTI)。XTI可形式化地描述为：*<term, docu-ID, elem-ID, start, end, position, level>*，其中*docu-ID*和*elem-ID*分别表示根据超级链接和文档内引用计算出来的d_w和r_w值；*start*和*end*分别为*term*所在的最近答案结点元素在XML文档树中的区间编码[8]，若元素u是元素v的祖先，则满足$start(u)<start(v) \wedge end(v)<end(u)$。因此，可根据答案结点$u$、$v$的*start*和*end*值来判断$u$、$v$之间的关系；*position*表示*term*在该答案结点中的位置，以便于多关键词的检索；*level*表示该答案结点在XML文档树中的深度。

3 搜索引擎的体系结构

搜索引擎的体系结构如图 3 所示。

XML 文档首先被全文扫描以获取它的结构和统计信息，生成扩展的词条索引。利用答案结点的区间编码可以得到答案结点间的祖先后裔关系，根据这个关系，还可以为用户获取新的信息进行导航。

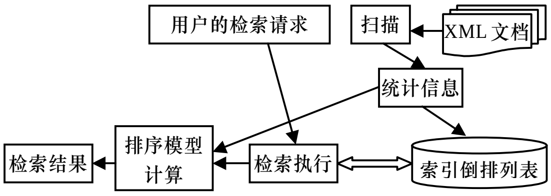

图 3 搜索引擎体系结构

4 结论

针对 XML 文档特征，本文分析了影响关键词检索结果排序的因素，主要包括检索结果的具体性、文档的链接和引用、关键词所在位置、答案结点的文本大小以及关键词之间的距离等；提出了关键词检索结果的排序模型，以及适合该模型的倒排列表索引结构和搜索引擎的体系结构。

与 XML 数据管理相关的工作，还有大量的研究有待我们去做，如本文主要讨论的是针对只与内容相关的关键词检索结果的排序，还有结合结构的（CAS）的关键词检索的研究，以及本文中未过多提及的答案结点、分词等的研究。

参考文献

1 Florescu D, Kossmann D, Manolescu I. Integrating Keyword Search into XML Query Processing[C]. Proc. of the Int'l. World Wide Web Conf., Amsterdam, Netherlands, 2000-05: 119-135.

2 Yates R B, Neto B R. Modern Information Retrieval[M]. ACM Press, 1999.

3 Guo L, Shao F, Botev C, et al. XRANK: Ranked Keyword Search over XML Documents[R]. Cornell University, 2003.

4 Liu S, Zou Q, Chu W W. Configurable Indexing and Ranking for XML Information Retrieval[C]. Proc. of the Int'l Conf. on SIGIR, Sheffield, South Yorkshire, UK, 2004: 88-95.

5 王晓玲. 面向 WEB 的 XML 数据管理技术研究[D]. 南京：东南大学, 2003.

6 万常选. XML 数据库技术[M]. 北京：清华大学出版社, 2005.

7 Xdanger. Google 的 PageRank 算法[Z]. 2003-12. http://blog.xdanger.com/ archives/2003/12/21/000053.html.

8 Clarke C L A, Cormack G V. Shortest-substring Retrieval and Ranking [J]. ACM Transaction on Information System, 2000, 18(1): 44-78.

~~~~~~

(上接第 39 页)

致系统无法满足全部到达的请求，这时控制机制就应该控制系统在一个合适的范围之内工作。该机制可以使系统资源能被合理分配、有效利用，在不可预测的情况下有效地控制系统负载，获得性能保证。

4 总结

本文提出基于反馈控制的通知服务接纳控制机制，实现通知服务的性能控制。仿真实验表明当到达的流量过高时，该机制可以使系统资源能被合理分配、有效利用，在不可预测的情况下有效地控制系统负载，获得性能保证。但是实际的模型往往是非线性的，用近似的线性化模型来建模是不准确的，将降低反馈控制的性能。对此，我们将进一步研究解决的问题。

参考文献

1 Shi Xiaoan, Zhou Xingshe, Wu Xiaojun, et al. Adaptive Control Based Dynamic Real-time Resource Management[C]. Proc. of IEEE International Conference on Machine Learning and Cybernetics, 2003: 3155-3159.

2 Object Mgmt. Group. Notification Service Specification (Version 1.2)[Z]. OMG Doc. formal/04-10-02 edition, 2004.

3 John A S, Tian H, Tarek F A, et al. Feedback Control Real-time Scheduling in Distributed Real-time Systems[C]. Proc. of the 22nd IEEE Real-time Systems Symposium, London, UK, 2001: 3-6.

4 Äström K J, Wittenmark B. Computer-controlled Systems-theory and Design[M]. NJ: Prentice Hall, Englewood Cliffs, 1990.

5 The Integration Server Company. OpenFusion Notification Service: Performance Evaluation[Z]. 2001.

6 张尧学, 方存好, 王　勇. 非精确计算中基于反馈的 CPU 在线调度算法[J]. 软件学报, 2004, 15(4): 616-623.
~~~~~~