计算机科学2004Vol. 31№. 9A

基于文档划分的XML信息检索研究

王晓燕　王海洋　洪晓光

（山东大学计算机科学与技术学院　济南250061）

摘　要　随着Internet的迅猛发展和XML的广泛应用，XML信息检索已成为网络检索技术的研究热点。本文对基于文档划分的XML信息检索技术进行了研究，利用XML的结构和语义信息，提出了一种能够针对具体的查询自行界定适于检索的信息单元的检索方法，来减少系统运行的计算开销，提高XML信息的检索速度。

关键词　XML，文档划分，信息检索

XML Information Retrieval Based-on Document Division

WANG Xiao-Yan　WANG Hai-Yang　HONG Xiao-Guang

(College of Computer Science and Technology, Shandong University, Jinan 250061)

Abstract　Along with the rapid development of Internet and the extensive use of XML, XML information retrieval techonology has become the focus of current research. It introduces XML information retrieval technology based on document division and presents a new method which can identify the suitable information units for retrieval automatically. Because adequately utilizing the context information in an XML document to fix on the granularity of information units, this method can reduce the cost of calculation and improve the efficiency of the information retrieval system.

Keywords　XML, Document division, Information retrieval

1　引言

随着网络技术的迅猛发展和信息量的骤然剧增，信息检索已经成为人们从Internet上获取有效信息不可或缺的技术手段，而当前XML的广泛应用更加速了信息检索技术的发展。作为一种结构化的文本文档，XML在物理形式上有着与其所表达的思想内容相对应的组织结构和层次关系，在逻辑表达上其语义信息同时存在于文档的文本与结构之中。信息检索系统可以根据XML文档中的元素标签来确定其结构，进而确定在文档中的哪一部分进行检索。

要进行信息检索，首先要界定对用户有意义的信息单元。传统的XML信息检索有两种：一种是以完整的XML文档为检索单元，检索结果是符合查询条件的文档集合。这种方法可以为用户提供全面的检索信息；但检索单元规模太大，结果中包含的与查询不相关的信息较多，会降低检索准确度并增加信息冗余。另一种是以XML元素为检索单元，检索结果是包含查询关键词密度最高的文档片段。这种方法可以提高检索精度；但由于XML元素数量巨大且尺寸偏小，检索性能因计算开销巨大会严重下降，而且返回信息相当琐碎，无法为用户提供其所需的完整信息。要实现基于关键词查询的XML信息检索，关键在于如何确定信息单元的粒度，以及用什么样的方法来计算信息单元与查询之间的相似度。本文对XML的信息检索技术进行了研究，利用文档自身携带的结构和语义信息，提出了一种基于文档划分的检索方法。这种方法可以针对具体的查询自动界定适于检索的信息单元，结合改进的向量空间模型计算信息单元与查询之间的相似度，从而减少系统计算开销，提高信息检索速度。

2　相关工作

为了实现Web中多元化的信息共享，使用户可以像使用现在的Web搜索引擎一样，无需指出文档结构，只要输入自己感兴趣的关键词，就可以自由地检索XML信息，本文的研究主要基于以下两个基础模型：XML文档的逻辑结构模型和向量空间模型。

2.1　XML文档的逻辑结构模型

在XPath数据模型[1]中，利用元素间的分层与包含关系，可以将一个XML文档转化为一个树型结构。一个XML文档的逻辑结构模型就是一棵以

王晓燕　硕士研究生，主要从事数据库工程与Web应用等方面的研究。王海洋　博士，教授，博士生导师，主要从事软件与数据库工程等方面的研究。洪晓光　博士，教授，主要从事数据库与网络应用等方面的研究。

文档节点为根节点、以文档中的元素节点为子节点、以节点间的父子关系来标识文档中的包含关系及附带属性的文档树，其中，节点间的父子关系由父结点指向子结点的有向边来标识。

为了对一棵 XML 文档树中的每一个节点进行标识，我们按照深度优先规则对其进行编号，其中根节点的编号为0。根据各节点到根节点的距离，对文档树进行层次划分。不妨设有父子关系的相邻节点间的路径权值为1，则节点到根节点间的路径权值总和即为该节点在文档树中所处的层数，到根节点有相同路径权值的节点为同层节点，其中根节点位于第0层，以根节点为父节点的节点位于第1层，以根节点为祖父节点的节点位于第2层……以此类推，得到一个 XML 文档树的层次划分模型，其中，节点所在的层数越高，其所处的位置越低。图1显示了一个 XML 文档树实例(仅以根节点和元素节点为例)，其中，节点 i 所标识的是 XML 文档中对应元素所标识的局部文档 D_i。

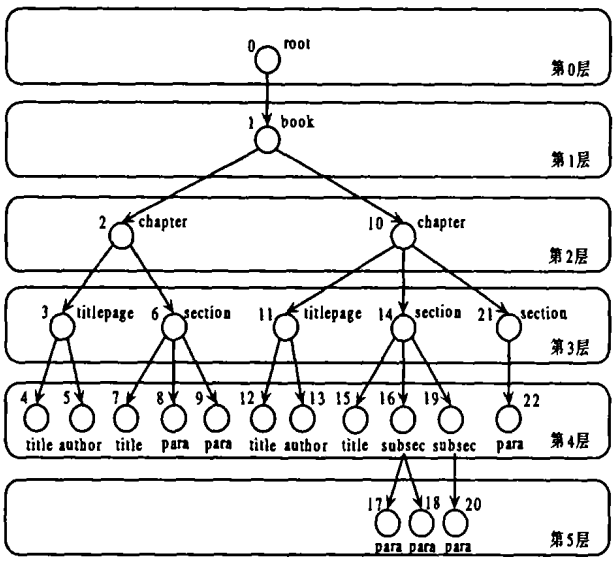

图1 一个 XML 文档树实例

2.2 向量空间模型

向量空间模型(Vector Space Model，VSM)[2]是信息检索领域中经典的计算模型，它采用向量空间模型对文档进行特征表达，根据一定的权值策略对特征项赋权，应用向量夹角余弦来度量文档与查询间的相似度，最后以相似度递减的顺序将检索结果提交给用户。

首先从文档 D 中提取其特征项组成文档的特征向量 $D(t_1,t_2,\cdots,t_n)$，其中 t_k 是特征项，$1\leqslant k\leqslant n$。根据特征项重要程度的不同，用附加权重 w_k 对文档 D 进行量化，表示为 $D(t_1,w_1;t_2,w_2;\cdots;t_n,w_n)$，在忽略特征项间的相关信息，即暂不考虑 t_k 在文档中的先后顺序并要求 t_k 互异时，可简记为 $D(w_1,w_2,\cdots,w_n)$，其中特征项 t_k 的权重为 w_k。这样，就可以用一个特征向量来表示一个文档，而两个文档 D_i 与 D_j 之间内容的相关程度可以用相似度 $Sim(D_i,D_j)$来度量。

3 基于文档划分的检索方法

要使用统一的检索技术进行基于关键词查询的 XML 信息检索，其基本思想是：首先获得以关键词方式提交的用户查询，在对文档标签进行查询匹配预处理的基础上，根据对 XML 文档的语义和结构分析进行文档划分，获得适于检索当前文档的信息单元(包括检索单元和检索结果单元)，根据向量空间模型进行信息单元与查询之间相似度的计算，最后建立倒排表，以相似度递减的顺序将检索结果单元提交给用户。目前要解决的问题是如何应用当前的信息检索技术来确定信息单元的粒度，以及用什么样的方法来计算信息单元与查询之间的相似度。

3.1 确定检索单元

尽管 Web 上的 XML 文档很多，然而其结构却千差万别，要从不同的 XML 文档中提取适于检索的信息单元是有一定难度的。一个 XML 文档是一个语义结构化的文档，其结构化不仅体现在文档元素标签存在的形式上，标签的名称更反映出该元素所描述的相关内容；由于一个 XML 文档可以被映射为一棵层次分明的文档树，文档树中具有父子关系的节点之间存在着以孩子节点来描述父亲节点的关系，也就是说，以某一节点为根节点的子树所标识的是一个与该节点名称相对应的知识领域，其子节点是对该领域相关信息的具体描述。基于上述分析，我们对检索单元的定义过程描述如下。

信息检索系统应用 XML 语法分析器对 XML 文档进行分析，构建出对应的文档树，并从中抽取出元素节点的标签内容；当有用户向系统提交查询时，系统首先将用户输入的关键词与抽取出的标签内容进行匹配：

(1)若匹配的节点含有非叶子节点，则找到所有匹配的节点中层次最高的节点，将其所在层次内的节点及出现在该层以上的所有叶节点作为该文档的检索单元；

(2)若匹配不成功或匹配成功的仅为叶子节点，则根据文档树的结构，确定含有元素节点数目最多的文档树层次，将其父层的每个节点以及父层以上的所有叶节点分别作为检索单元。

3.2 计算检索单元与查询之间的相似度

对于得到的每一个检索单元，提取其所含有的所有节点、文本及其串值作为文本内容，根据一定的权值策略生成该文档的文档向量。对输入到系统中的每一个查询，产生一个查询向量。通过计算检索单元与查询向量之间夹角的余弦，得到二者之间的相

似度。

设节点 i 标识一个检索单元。信息检索系统计算在所有的 XML 文档中出现的特征项 $t_k(k=1,2,\cdots,n)$的频度，应用2.2中的向量空间模型理论生成检索单元 D_i 的特征向量为 $D_i=(w_{i1},w_{i2},\cdots,w_{in})$，其中，特征项权值 $w_{ik}(k=1,2,\cdots,n)$是指检索单元 D_i 中第 k 个关键词 t_k 的权重，按照特征项频率 tf_{ik} 和反比文献频率 idf_k 的 TF·IDF 方法来计算：

$$w_{ik}=tf_{ik}\times\log(N/n_k+0.01)/\sqrt{\sum_{k=1}^{n}[tf_{ik}\times\log(N/n_k+0.01)]^2} \quad (1)$$

其中，tf_{ik}表示查询关键词 t_k 在 D_i 中出现的频率；$\log(N/n_k+0.01)$是 t_k 在检索单元集合中所有分布情况的量化(其中，N 表示检索单元的数目，n_k 表示出现过 t_k 的检索单元数目)；分母是对各分量的标准化，从而使每一个 t_k 的权重在[0,1]之间。

当用户输入查询关键词时，信息检索系统自动产生一个查询向量 Q。由于查询向量 Q 的基数与 D_i 相同，并且其元素值并不依赖于关键词是否包含在查询关键词中，因此我们可以将查询特性向量表示为 $Q=(q_1,q_2,\cdots,q_n)$。如果 t_k 在查询关键词中，则 q_k 为1；否则，q_k 为0。

检索单元 D_i 与查询 Q 之间的相似度以两向量之间夹角的余弦值来度量。该余弦值越大则说明二者之间的夹角越小，进而说明检索单元 D_i 与查询 Q 之间的相似度越高。其计算公式为：

$$Sim(D_i,Q)=\cos\theta=\sum_{k=1}^{n}w_{ik}\times q_k/\sqrt{(\sum_{k=1}^{n}w_{ik}^2)(\sum_{k=1}^{n}q_k^2)} \quad (2)$$

3.3 确定检索结果单元

为了提交给用户大小适中、语义完全的检索信息，根据相似度在 XML 文档结构中的映射关系，我们定义一个 XML 文档的检索结果单元为：

(1)对于检索单元定义过程中(1)的情况，其检索结果单元为检索单元本身；

(2)对于检索单元定义过程中(2)的情况，其检索结果单元为检索单元到文档根节点的路径上，同层节点中含有相同元素标签名称的最低结构层内的所有节点(不包含检索单元层节点)；如果这样的节点不存在，则定义文档根节点为检索结果单元。

3.4 计算检索结果单元与查询之间的相似度

为了将检索单元 D_i 与查询 Q 之间的相似度传递到检索结果单元中，对于 XML 文档中被确定为检索结果单元的 n 号节点，通过下式计算其与查询 Q 之间的相似度：

$$Sim(D_n,Q)=f(Sim(D_{c1},Q),Sim(D_{c2},Q),\cdots,Sim(D_{cm},Q)) \quad (3)$$

其中，$c_1,c_2,\cdots,c_m$ 为 n 号节点的孩子节点，$Sim(D_{ck},Q)$表示根节点编号为 c_k 的子树与查询之间的相似度，其中$1\leqslant k\leqslant m$。本文中采用的 $f()$为 p 范数函数($p=2$)[3]：

$$f(s_1,s_2,\cdots,s_m)=1-\left(\frac{(1-s_1)^p+(1-s_2)^p+\cdots+(1-s_m)^p}{m}\right)^{1/p} \quad (4)$$

举例来说，当用户输入查询关键词“信息检索”时，对于图1所示的文档，系统首先将抽取出的元素标签与查询关键词进行比较，未发现匹配成功者，按照检索单元定义(2)的情形进行处理：确定含有元素节点最多的层次是文档树中的第4层，将其父层内的每一个节点确定为一个检索单元，得到该文档的检索单元为文档树中编号为3、6、11、14、21的节点所标识的局部文档；根据式(2)，计算检索单元 D_3、D_6、D_{11}、D_{14}和 D_{21}与查询之间的相似度；按照检索结果单元定义(2)的情形，确定该文档的检索结果单元为编号为2、10的节点所标识的局部文档；根据公式(3)和(4)，计算检索结果单元 D_2和 D_{10}与查询 Q 之间的相似度分别为：$Sim(D_2,Q)=f(Sim(D_3,Q),Sim(D_6,Q))$；$Sim(D_{10},Q)=f(Sim(D_{11},Q),Sim(D_{14},Q),Sim(D_{21},Q))$。其中，2号节点是3号和6号节点的父节点，10号节点是11号、14号和21号节点的父节点，若检索结果单元与检索单元之间为祖先/子孙关系，则根据式(4)层层向上推导，直到得到检索结果单元与查询之间的相似度为止，最后按照相似度的递减顺序将检索结果提交给用户。

应用上述方法可以从 XML 文档中自动提取适于检索的信息单元。由于以检索节点为根节点的子树中至少含有一个叶节点，因此，检索单元的数目不会超过(一般会大大少于)叶节点的数目，并且其规模介于整个文档与文档元素之间；由于每个检索结果单元至少含有一个检索单元，因此，检索结果单元的数目不会大于检索单元的数目且大小适中。这样就解决了传统的 XML 方法中信息单元数量过大以及信息冗余或不足的问题。

对于一个给定的 XML 文档，信息单元的粒度只需计算一次，因此对每次检索的速度真正产生影响的是检索单元 D_i 与查询 Q 之间相似度的计算。在以前的方法中，检索单元 D_i 与查询 Q 之间的相似度必须从叶节点算起：由式(2)得到每个叶节点与查询 Q 之间的相似度，通过式(3)计算上层节点与查询 Q 之间的相似度；而在我们提出的方法中，由于针对具体的查询定义了合适的信息单元，相似度的计算仅在信息单元中进行：通过式(2)计算检索单元与查询 Q 之间的相似度，通过式(3)计算检索结果单元与查询 Q 之间的相似度。由于信息单元的数

(下转第138页)

如果查询为:查询教"面向对象技术"课程的教授的姓名。那么本文可以形成全局查询为:

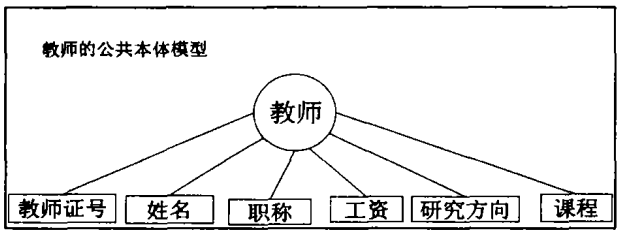

图5 教师的公共本体模型

select 姓名 from 教师 where 课程="面向对象技术"and 职称="教授"。先把该查询发送到缓存管理器,如果缓存中有该查询的结果并且结果符合用户要求,则由缓存管理器直接把结果返回给用户界面,查询过程结束。否则通过查询分解,假设在系统中只有财务处与教务处与教师相关,那么可以分解查询如下:select 姓名 from 教师 where 职称="教授";select 姓名 from 教师 where 课程="面向对象技术",并且这两个子查询查询出的作为主键的教师证号是相同的。通过从公共本体与各领域本体映射表中查找各项的对应项,从而得到两个局部子查询为:select 姓名 from 财务处 where 职称="教授";select 姓名 from 教务处 where 课程="面向对象技术",同时分别得到这两个局部子查询的地址信息:com. finacial. sdu. edu. ,com. dean. sdu. edu,然后查询分发器根据地址去调用相应子查询的数据源在Web服务器上部署的方法,然后得到返回的查询结果。如果结果需要合并转换,再由结果收集器合并结果并根据用户查询进行转换,然后显示给用户。该查询结果还提交给缓存管理器,缓存管理器根据算法3.1决定是否将结果存入缓存库,是否需要合并存入缓存库。到此查询完成。

结束语 本文提出了处理Web查询的一个总体框架,并对缩短查询的响应时间提出了解决办法,但框架中缓存库中存放的信息并不是实时更新的,如何保证缓存库中的信息的正确性,虽然本文提出了可信度限制,但并不能保证信息的实时更新,如何让缓存数据库中存放的信息自动实时更新,并计算维护的成本是以后准备研究的工作。

参 考 文 献

1 Braga R M M, Werner C M L. Using ontologies for Domains Information Retrieval. Computer Science Department Federal University of Rio de Janeiro, 0-7695-0680-1/00 (c)2000 IEEE

2 Cruz I F, Rajendran A. Semantic Data Integration in Hierarchical Domains. University of Illinois at Chicago, 1094-71671031©2003 IEEE

3 Ashish N, Knoblock C A, Shahabi C. Selectively Materializing Data in Mediators by Analyzing User Queries. Integrated Media Systems Center. Information Sciences Institute and Department of Computer Science University of Southern California, 4676 AdmiraltyWay, Marina del Rey, CA 90292

4 邓志鸿,唐世渭,杨冬青. 面向语义集成——本体在Web信息集成中的研究进展. 北京大学计算机科学与技术系. 计算机应用, 2002(1)

5 邓志鸿,唐世渭,张铭,杨冬青,陈捷. Ontology研究综述. 北京大学计算机科学与技术系. 北京大学学报(自然科学版), 2002(5)

(上接第104页)

目在一般情况下要远远少于叶节点的数目,因此可以避免信息单元与查询之间过多的相似度计算量,可以在提交给用户大小适中、语义完全的检索结果信息的同时,加速信息检索的运行效率,提高用户对检索结果的满意程度。

结论 本文对基于文档划分的XML信息检索技术进行了研究,提出了一种用统一的检索技术进行基于关键词查询的XML信息检索方法。由于充分利用了XML文档的结构和语义特点,比较传统的XML信息检索技术,该方法可以在XML文档中自动界定适于检索的信息单元,以减少系统运行的计算开销,提高信息检索速度。在今后的研究中,将对这种方法进行进一步完善和改进,在不影响查询速度的前提下,研究提高其查全率和查准率,进而提高检索的综合性能。

参 考 文 献

1 Berglund A, Boag S, Chamberlin D, et al. XML Path Language (XPath)2.0. W3C Working Draft. World Wide Web Consortium (W3C), May 2003. http://www.w3.org/TR/xpath20/

2 Salton G, Wong A, Yang C S. A vector space model for automatic indexing. Communications of the ACM. New York, USA, 1975

3 Lee J. Analyzig the Effectiveness of Extended Boolean Models in Information Retrieval. In: Proc. of ACM SIGIR'94. Dublin, Ireland, 1994

4 Hatano K, Kinutani H, Yoshikawa M, Uemura S. Information Retrieval System for XML Documents. In: Proc. of the 13th Intl. Conf. on Database and Expert Systems Applications. Aix-En-Province, France, 2002

5 Hayashi Y, Tomita J, et al. Searching text-rich XML documents with relevance ranking. ACM SIGIR 2000 Workshop on XML and Information Retrieval. Athens, Greece, 2000