作为 XML 文档的近似搜索的基础,首先要能够准确地度量查询与文档、文档与文档间的相关(似)性.传统的信息检索技术利用向量空间模型来表示一个文档,并利用代表文档的空间向量间的距离来度量两个文档间的相关程度.尽管我们也可以使用类似的方法来计算 XML 文档间的相似程度,但向量空间模型无法反映 XML 文档中的元素嵌套结构的语义.一般地,一个 XML 文档可以模型化为一棵树或一个图,两个 XML 文档间的相似度可以用这两棵树(图)间的距离来度量.在 XML 出现之前,已有许多工作[6~10]研究了两棵树(图)间的相似测度的问题,其中最自然和应用最广的测度是树的编辑距离.Tai[6]最早提出了利用编辑距离来度量两棵树(图)间的差异.在 Tai 的工作的基础上,Zhang 和 Shasha[7~9]等提出了计算两棵树间的各种编辑距离的算法.

这些工作研究的是一般意义上的树间距离.XML 文档本身具有许多显著的特点.我们认为,在度量两个 XML 文档间的距离时应充分考虑这些特点.同时,还应注意到,Web 上的 XML 文档往往是大规模的,在处理海量的 XML 文档时,要遍历每棵文档树中的每个结点在很多情况下是不现实的.

针对上述问题,本文提出了一个适于定量度量 XML 文档间的相似度的新的编辑距离(称为 XED 距离).利用结点间的模拟关系,一个 XML 文档可以表示为一棵精简的、带权重的结构索引树.本文从理论上严格证明了两个 XML 文档间的距离可以通过计算它们的索引树间的距离来测定.由于索引树通常远小于原文档树结构的大小,利用索引树可以大大提高判定两个 XML 文档结构相似度的效率.

本文的研究动因来自于 XML 文档近似检索[11,12],但本文考虑的是广泛意义上的 XML 文档的相似度计算问题,因而具有较强的通用性.本文的研究课题还可以用于 XML 文档聚类、XML 文档结构抽取、XML 文档的变换检测[13]以及 XML 视图的增量计算和维护等方面.

2　引　例

XML 文档的 DOM 树结构可以看作是该文档的树型结构表示.为便于描述,本文在这种树型结构中不特别区分元素和属性,也不考虑树中的叶结点的文本内容.同时,在许多实际应用中,只有结点的嵌套关系才是重要的,而同一元素下的子元素或属性出现的次序是不重要的①.因此,本文使用无序的树型结构来表示 XML 文档,形式定义如下.

定义 1(XML 文档树).　一个 XML 文档是一个二元组(N,E).其中

(1)N 是结点集,每个结点 $n\in N$ 对应文档中的一个元素或属性,包含一个结点对象标识 Nid(按深度优先排序),结点对应的元素(属性)名 *name* 和表示元素(属性)中字符串的内容 *content*.

(2)$E\subseteq N\times N$ 是有向边集合.每条边 $e\in E$ 代表结点间的嵌套关系.

在以后的讨论中,我们也将结点名 *name* 和边 E 看作是一个函数,即 $name(n)$表示结点 n 的名字,$E(n)$表示结点 n 的子结点集,同时,用 $E^{-1}(n)$表示 n 的父结点.

图 1 给出了两个来自不同 Web 站点的关于旅馆信息的 XML 文档树的例子.从图 1 中我们可以看出,两个 XML 文档的结构相似的情况可能十分复杂.同样的子树结构可能出现在不同的层次上(如 *address* 子树).由于 XML 文档中可能存在大量的重复,一个文档中的部分可能与另一个文档中的不同的部分相似.如图中左边的文档只有一个 *rooms* 元素,其下边有多种类型的房价信息,而右边的文档有多个 *rooms*,每个 *rooms* 元素仅包含一种类型的房价.此外,即使子树结构一样,二者的数量也可能不同,如:一个旅馆可能有 10 种类型(*type* 结点)的房间,而另一个旅馆则可能仅有 3 种类型的房间.

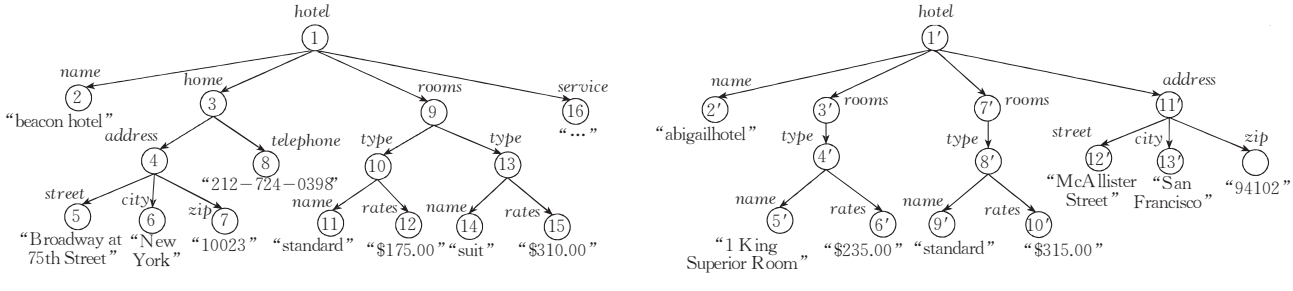

图 1　两个 XML 文档的 XML 文档树结构

① 实际上,计算两个有序的文档的相似度比计算两个无序的文档的相似度要更加容易.

3 XML 编辑距离

本节先简要介绍树间的编辑距离的概念,再在此基础上提出 XML 编辑距离.

Tai[6]最早使用编辑距离来计算两棵树间的相似度.其基本思想是将两棵树间的距离定义为利用编辑操作将一棵树转化为另一棵所需的代价.以下三种操作称为编辑操作,简称 ED 操作:

(1)$Update(u,v)$.将结点更新为 v;

(2)$Delete(u)$.将结点 u 的所有儿子作为 u 的兄弟结点插入到 u 的父亲结点的子树,并删除 u;

(3)$Insert(x,u)$.在结点 x 下插入结点 u 并将 x 的部分儿子作为 u 的儿子结点.

每个 ED 操作对应一个代价.令 γ 为一个代价函数,它为每个操作赋予一个非负整数代价值,记为 $\gamma(u\to v)$.γ 满足如下条件:

(1) $\gamma(u\to v)\geq 0,\gamma(u\to u)=0$;

(2) $\gamma(u\to v)=\gamma(v\to u)$;

(3) $\gamma(u\to v)+\gamma(v\to w)\geq\gamma(u\to w)$.

为便于说明,假定所有的 ED 操作的代价都为单位代价 1.令 S 为从 T_1 到 T_2 的 ED 操作系列 $s_1,s_2,\cdots,s_n$,则 S 的代价是这些 ED 操作的代价的总和,即 $\gamma(S)=\sum_{i=1}^{|S|}r(s_i)$.

根据文献[6],两棵树 T_1 和 T_2 间的编辑距离定义为将 T_1(或 T_2)转化为 T_2(或 T_1)的最小的操作序列的代价.

定义 2(ED 距离).

$ED(T_1,T_2)=\min_S\{\gamma(S)\mid S$ 是从 T_1 到 T_2 的 ED 操作系列$\}$.

一个从 T_1 到 T_2 的 ED 操作系列 S 将 T_1 中的部分结点转化为 T_2 中的部分结点.其效果相当于一个从 T_1 到 T_2 的映射.这样的映射称为 ED 映射,定义如下.

定义 3(ED 映射).给定两棵无序树 T_1 和 T_2,一个从 T_1 到 T_2 的 ED 映射是一个三元组 (M,T_1,T_2).其中,M 是一个有序结点对 (u,v) 构成的集合,满足:

(1) u 是 T_1 中的结点,v 是 T_2 中的结点;

(2) 对 M 中的任意两个结点对 (u_1,v_1) 和 (u_2,v_2),满足

(a) $u_1=u_2$ 当且仅当 $v_1=v_2$;

(b) u_1 是 u_2 的祖先,当且仅当 v_1 是 v_2 的祖先;

从 T_1 到 T_2 的 ED 映射 M 的代价函数为

$$\gamma(M)=\sum_{(u,v)\in M}\gamma(u\to v)+\sum_{\{u\mid\neg(\exists v,(u,v)\in M)\}}\gamma(u\to\Lambda)+\sum_{\{v\mid\neg(\exists u,(u,v)\in M)\}}\gamma(\Lambda\to v).$$

以下引理描述了从 T_1 到 T_2 的 ED 映射 M 与 T_1 到 T_2 的 ED 操作序列 S 之间的关系,其证明见文献[6].

引理 1. 任意给定一个从 T_1 到 T_2 的 ED 操作序列 S,都存在一个从 T_1 到 T_2 的 ED 映射 M 满足 $\gamma(M)\leq\gamma(S)$.反之,任给定一个从 T_1 到 T_2 的 ED 映射 M,都存在一个从 T_1 到 T_2 的 ED 操作序列 S 满足 $\gamma(S)=\gamma(M)$.

根据该引理,一个从 T_1 到 T_2 的编辑距离(以下称为 ED 距离)可以用它们的 ED 映射 M 描述如下.

定理 1. $ED(T_1,T_2)=\min_M\{\gamma(M)\mid M$ 是从 T_1 到 T_2 的 ED 映射$\}$.

例 1. 图 2 给出了图 1 中的文档树 T_1 和 T_2 之间的一个 ED 映射(如虚线所示).与该映射对应的操作系列为 $S=\{1\to1',2\to2',3\to\Lambda,4\to11',5\to12',6\to13',7\to14',8\to\Lambda,9\to3',10\to4',11\to5',12\to6',13\to8',14\to9',15\to10',16\to\Lambda,\Lambda\to5'\}$,其代价总和 $\gamma(M)=\gamma(S)=4$.该映射同时也是从 T_1 到 T_2 的代价最小的 ED 映射.因此,$ED(T_1,T_2)=4$.

ED 映射能够精确地度量两棵树间的差异,然而,Zhang 和 Shasha 等在文献[8]中证明了求两棵无序树间的最小映射问题是 NP-完全的.文献[9]

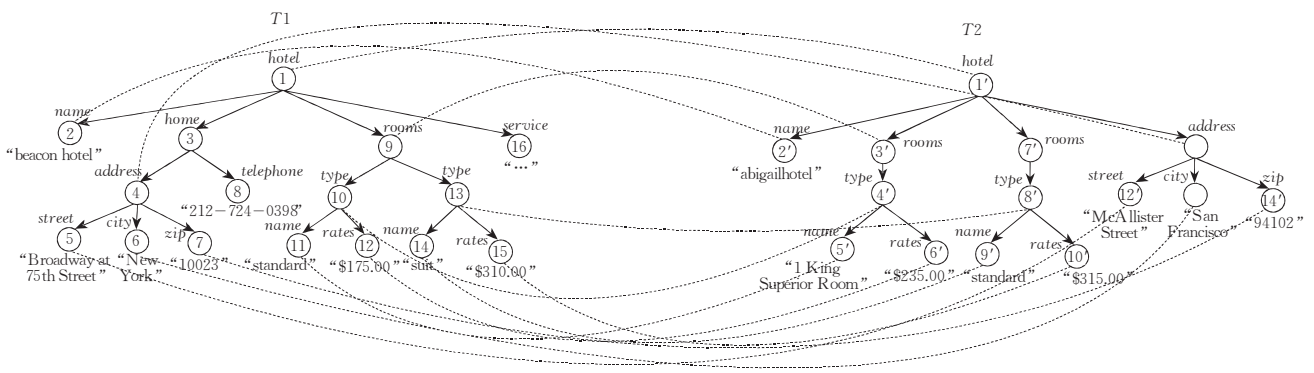

图 2 树 T_1,T_2 之间的 ED 映射

提出了一个求两棵无序树间的受限 ED 距离的多项式时间的算法.然而,即使是多项式时间的算法也要遍历两棵树中的每一个结点,这在处理大量的 XML 文档时代价仍十分昂贵.实际上,ED 距离定义的是一般意义上的树间距离,而 XML 文档中的标记具有一定的语义信息,且存在大量的重复.根据 XML 文档的特点,我们可以极大地缩小搜索空间.例如:对图 1 引例中的两个 *hotel* 文档,我们不必比较不同名的结点,也无需将左边两个不同的 *type* 结点下的 *name* 结点分别与右图中的 *hotel* 下的 *name* 和 *type* 下的 *name* 作比较.

基于以上观察,本文提出了一个适于 XML 文档的新的编辑距离,称为 XED 距离.在给出该定义之前,我们先引入两个概念:两个结点的最小共同祖先和结点的签名(signature).

定义 4(最小共同祖先).　令 u 和 u' 为 XML 树 T 中的两个结点,若结点 m 是 u 和 u' 的公共祖先,且在 T 中不存在结点 $n\neq m$ 满足 n 是 u 和 u' 的公共祖先,且 n 是 m 的子孙结点,则我们称 m 是 u 和 u' 的最小公共祖先,记作 $m=lca(u,u')$.

显然,XML 文档树中任两个结点的最小共同祖先 lca 一定存在.

定义 5(结点签名).　令 u 和 u' 为文档树 T 中的两个结点,u' 是 u 的祖先,从 u 到 u' 的路径上的标记序列 $name(u').name(x_1)\cdots.name(x_n).name(u)$ 称为 u' 到 u 的签名,记作 $signature(u',u)$.

现在,我们定义两棵 XML 树间的 XED 映射.

定义 6(XED 映射).　给定两棵 XML 树 T_1 和 T_2,一个从 T_1 到 T_2 的 XED 映射是一个三元组 (M,T_1,T_2).其中,M 是一个有序结点对 (u,v) 构成的集合,满足:

(1) M 是一个 ED 映射;

(2) 任意一个结点对 $(u,v)\in M$,$name(u)=name(v)$;

(3) M 中的任意两个结点对 (u_1,v_1) 和 (u_2,v_2),$u_1,u_2,v_1,v_2\in\{\sum-\Lambda\}$,满足:

(a) $signature(lca(u_1,u_2),u_1)=signature(lca(u_1,u_2),u_2)$ 当且仅当 $signature(lca(v_1,v_2),v_1)=signature(lca(v_1,v_2),v_2)$;

(b) 对 M 中的另一结点对 (u_3,v_3),若 $lca(u_1,u_2)$ 是 u_3 的祖先,当且仅当 $lca(v_1,v_2)$ 是 v_3 的祖先①.

我们将与 XED 映射对应的操作称为 XED 操作.引理 1 对 XED 映射和 XED 操作同样成立.同样地,两棵树间的 XED 距离具有如下性质.

定理 2.　$XED(T_1,T_2)=\min_M\{\gamma(M)\mid M$ 是从 T_1 到 T_2 的 XED 映射$\}$.

定理 3.　XED 距离可以在多项式时间内计算.

证明.　容易证明,XED 距离属于文献[12]中提出的受限的 ED 距离,文献[12]证明了两棵树间的受限的 ED 距离可以简化为网络最小代价最大流问题,时间复杂度为 $O(|T_1|\times|T_2|\times(\deg(T_1)+\deg(T_1)\times\log_2(\deg(T_1)+\deg(T_1))))$.因此,XED 距离也可以在多项式时间内计算②.

例 2.　图 2 中的两棵 XML 树 T_1 和 T_2 之间的 ED 距离为 4,XED 距离也为 4.代价最小的 XED 映射如图 2 中的虚线所示.从 XED 的定义可知,XED 映射比 ED 映射有更多的限制,XED 距离可能大于 ED 距离(读者容易举出这样的例子).

4　结构索引

如前所述,XML 文档中的标记可能存在大量的重复.如:一个描述图书馆书籍的 XML 文档可能包含大量的 *book* 元素,而每本书又有多个 *author* 子元素.信息检索技术将一个文档表示为一个关键字的权重向量,并用两个文档向量间的距离来度量两个文档间的相似度.采用类似的思想,我们可以将一个 XML 文档的结构用一棵带权重的树来表示.

我们利用结点间的"模拟"(simulation)[14]关系来构造一个 XML 文档的结构索引.结点间的模拟关系定义如下.

定义 7(结点模拟关系).　两个结点 u_1 和 u_2 具有模拟关系,记作 $u_1\approx u_2$,当且仅当 u_1 和 u_2 满足如下条件:

(1) $name(u_1)=name(u_2)$;

(2) 若 u_1 是根结点,则 u_2 也是根结点;

(3) 否则,$E^{-1}(u_1)\approx E^{-1}(u_2)$.

结点模拟关系是一种等价关系.它将一棵树中的结点划分为不同的等价类,每个等价类中的任意两个结点都具有模拟关系.在后边的讨论中,我们用 $[u]$ 表示结点 u 所属的等价类.根据结点间的模拟关系,我们可以将一棵 XML 树 T 精简为一棵带权重的树(简称为权重树)I,称为 T 的结构索引.结构索引 I 的构造性定义如下.

定义 8(结构索引).　一棵 XML 树 T 的结构索引 I 是一棵无序树,I 中的每个结点 u_I 表示 T 中的一个等价类集合,且带有一个权重因子 w,满足:

(1) I 中的每一个结点在 T 中都有一个结点等

① 注意这里的条件(b)与定义 3 中的条件(b)不同:这里的 $lca(u_1,u_2)$ 和 $lca(v_1,v_2)$ 不一定要出现在 M 中.

② 这里,$|T|$ 表示 T 中的结点数,$\deg(T)$ 表示 T 中结点的最大出度.

价类与之对应,且 T 中的每个结点等价类在 I 中都有唯一的一个结点与之对应.

(2) I 中的每条边(u_I,v_I)在 T 中都至少有一条边(u,v)与之对应,满足 $u_I=[u]$,$v_I=[v]$,且 T 中的每条边(u,v)在 I 中都有唯一的一条边(u_I,v_I)与之对应,满足$[u]=u_I$,$[v]=v_I$.

I 中的每个结点的权重的取值可能有多种选择.在后边的讨论中,我们取 $w=|u_I|$,并简单地将 w 看作一个函数,即 $w(u_I)$表示 u_I 的权重. Tarjan 在文献[15]中提出了一个计算图中结点间的等价关系的 $O(m\log n)$时间复杂度的算法(其中 m 表示边数,n 表示结点数).利用该算法,我们可以在$O(m\log n)$时间内构造一棵 XML 树 T 的结构索引 I.

前面提到的三种编辑操作同样适用于结构索引.但是,树 T 的结构索引 I 中的结点表示的是 T 中的一个等价类集合,因此,编辑操作的代价与 T 中的代价不同.我们提出树 T 的结构索引 I 上的 XED 操作的代价模型如下:

(1) If $name(u_I)=name(v_I)$ then $\gamma(u_I\to v_I)=|w(u_I)-w(v_I)|$,else $\gamma(u_I\to v_I)=w(u_I)+w(v_I)$;

(2) $\gamma(u_I\to\Lambda)=w(u_I)$;

(3) $\gamma(\Lambda\to v_I)=w(v_I)$.

于是,两个结构索引 I_1 到 I_2 的 XED 映射 M_I 的代价如下:

$$\gamma(M_I)=\sum_{(u_I,v_I)\in M_I}\gamma(u_I\to v_I)+\sum_{\{u_I|\neg(\exists v_I,(u_I,v_I)\in M_I)\}}\gamma(u_I\to\Lambda)+\sum_{\{v_I|\neg(\exists u_I,(u_I,v_I)\in M_I)\}}\gamma(\Lambda\to v_I).$$

显然,引理 1 对结构索引树仍然成立.结构索引 I_1 和 I_2 间的 XED 距离计算如下:

$$XED(I_1,I_2)=\min_{M_I}\{\gamma(M_I)\mid M_I\text{ 是从 }I_1\text{ 到 }I_2\text{ 的 XED 映射}\}.$$

下边的引理说明同一棵树中等价的结点在 XED 映射下也等价.

引理 2. 若 M 为 XML 文档树 T_1 到 T_2 的 XED 映射,给定 M 中的任意两个结点对(u_1,v_1)和(u_2,v_2),$u_1\approx u_2$ 当且仅当 $v_1\approx v_2$.

证明. 我们先证明若 $u_1\approx u_2$,则 $v_1\approx v_2$ 成立.

设 $u=lca(u_1,u_2)$,$v=lca(v_1,v_2)$,u 到 u_1 和 u_2 的路径分别为 $u.x_1\cdots x_k.u_1$ 和 $u.y_1\cdots y_h.u_2$,则我们有如下观察:(1) $k=h$ 和(2)对所有的 $1\le i\le k$,都有 $name(x_i)=name(y_i)$成立.假设(1)不成立,不妨设 $k>h$,由于 $u_1\approx u_2$,根据$\approx$关系的定义,有 $y_h\approx x_k$,$y_{h-1}\approx x_{k-1}$,$\cdots$,$y_1\approx x_{k-h-1}$,$u\approx x_{k-h}$.由于 u 是 u_1 和 u_2 的 lca,因此,$u=x_{k-h}$,即 $k-h=0$.再根据$\approx$的定义,$y_i\approx x_i$ 对所有的 $1\le i\le n$ 都成立,故对所有的 $1\le i\le n$,$name(y_i)=name(x_i)$.综合(1)和(2),我们有 $signature(u,u_1)=signature(u,u_2)$.于是,由 XED 映射的定义,$signature(v,v_1)=signature(v,v_2)$成立.再设 v 到 v_1 和 v_2 的路径分别为 $v.x_1\cdots x_m.v_1$ 和 $v.y_1\cdots y_n.v_2$,则同样有 $m=n$ 和对所有的 $1\le i\le m$,都有 $name(x_i)=name(y_i)$,且 $name(v_1)=name(v_2)$成立.由于 $v\approx v$,$x_1\approx y_1$,$x_2\approx y_2$,$\cdots$,$x_m\approx y_m$,因此,$v_1\approx v_2$ 也成立.

类似地,我们可以证明若 $v_1\approx v_2$,则 $u_1\approx u_2$.

证毕.

引理 3. 任意给定一个从 I_1 到 I_2 的 XED 操作系列 S_I,都存在一个从 T_1 到 T_2 的 XED 操作系列 S 与之对应,且 $\gamma(S)=\gamma(S_I)$.反之,任意给定从 T_1 到 T_2 的 XED 操作系列 S,都存在一个从 I_1 到 I_2 的 XED 操作系列 S_I 与之对应,且 $\gamma(S_I)=\gamma(S)$.

证明. (1)对引理的前半部分,设 u_I 和 v_I 对应的等价类集合分别为$\{u_1,u_2,\cdots,u_{w(u_I)}\}$和$\{v_1,v_2,\cdots,v_{w(v_I)}\}$,$k=\min(w(u_I),w(v_I))$,我们对从 I_1 到 I_2 的 XED 操作系列 S_I 中的每一个操作 s_I 按操作类型归纳证明:存在一个从 T_1 到 T_2 的 XED 操作 S_j 与之对应(设 S_j 初始时为空):

情形 1:若 s_I 为操作 $u_I\to v_I$ 且 $name(u_I)=name(v_I)$,$\gamma(s_I)=|w(u_I)-w(v_I)|$,则我们将操作系列 $u_1\to v_1$,$u_2\to v_2$,$\cdots$,$u_k\to v_k$ 加入 S_j 中.此外,若 $w(u_I)>w(v_I)$,则将 u_I 对应的等价类集合中的剩余结点 u_{k+1},u_{k+2},$\cdots$,$u_{w(u_I)}$ 从 T_1 中删除,即将操作系列 $u_{k+1}\to\Lambda$,$u_{k+2}\to\Lambda$,$\cdots$,$u_{w(u_I)}\to\Lambda$ 加入 S_j.否则,若 $w(u_I)<w(v_I)$,则将 v_I 对应的等价类集合中剩余的结点 v_{k+1},v_{k+2},$\cdots$,$v_{w(v_I)}$ 插入 T_1,即将操作系列 $\Lambda\to v_{k+1}$,$\Lambda\to v_{k+2}$,$\cdots$,$\Lambda\to v_{w(v_I)}$ 加入 S_j.于是,

$$\gamma(S_j)=\begin{cases}\sum_{i=1}^{k}r(u_i\to v_i)+\sum_{i=k+1}^{|U_I|}r(u_i\to\Lambda), & w(u_I)\ge w(v_I)\\ \sum_{i=1}^{k}r(u_i\to v_i)+\sum_{i=k+1}^{|V_I|}r(\Lambda\to v_i), & \text{否则}\end{cases}$$

由于 $name(u_I)=name(v_I)$,$\sum_{i=1}^{k}r(u_i\to v_i)=0$,上边的函数可以统一为$|w(u_I)-w(v_I)|$,因此,$\gamma(S_j)=\gamma(s_I)$成立.

情形 2:若 s_I 为操作 $u_I\to v_I$ 且 $name(u_I)\neq name(v_I)$,则将操作系列 $u_1\to\Lambda$,$\cdots$,$u_{w(u_I)}\to\Lambda$,$\Lambda\to v_1$,$\Lambda\to v_2$,$\cdots$,$\Lambda\to v_{w(v_I)}$ 加入 S_j,此时,$\gamma(S_j)=w(u_I)+w(v_I)=\gamma(s_I)$.

情形 3:若 s_I 为 $u_I\to\Lambda$,则将 u_I 中所有的结点 u_1,u_2,$\cdots$,$u_{w(u_I)}$ 从 T_1 中删除,即将操作系列 $u_1\to\Lambda$,$u_2\to\Lambda$,$\cdots$,$u_{w(u_I)}\to\Lambda$ 加入 S_j.S_j 的代价 $\gamma(S_j)=w(u_I)=\gamma(s_I)$.

情形 4:若 $\Lambda\rightarrow v_I$,则将 v_I 中的结点 v_{k+1},v_{k+2},…,$v_{w(v_I)}$ 插入 T_2,即将操作系列 $\Lambda\rightarrow v_1$,$\Lambda\rightarrow v_2$,…,$\Lambda\rightarrow v_{w(v_I)}$ 加入 S_j. 仍有 $\gamma(S_j)=w(v_I)=\gamma(s_I)$.

综合上边的情形 1～情形 4 的证明可得:与 M_I 对应的每一个 XED 操作系列 S_I 都有一个从 T_1 到 T_2 的 XED 操作系列 S 与之对应,且 $\gamma(S_I)=\gamma(S)$.

(2)下面我们证明任意一个从 T_1 到 T_2 的 XED 操作系列 S 都有一个从 I_1 到 I_2 的 XED 操作系列 S_I 与之对应,且 $\gamma(S_I)=\gamma(S)$.

设操作系列 S 为 s_1,s_2,…,s_n,我们先将 s_1,s_2,…,s_n 按它们的操作数划分为 k 个子集 S_1,S_2,…,S_k. 其中每个子集 S_j 中的任意两个操作 $u_{j1}\rightarrow v_{j1}$ 和 $u_{j2}\rightarrow v_{j2}$ 都满足如下 3 个条件之一:(1) $u_{j1}\approx u_{j2}$ 且 $v_{j1}\approx v_{j2}$;或(2) $u_{j1}=u_{j2}=\Lambda$ 且 $v_{j1}\approx v_{j2}$;或(3) $u_{j1}\approx u_{j2}$ 且 $v_{j1}=v_{j2}=\Lambda$;设满足条件(1),(2)和(3)的集合的大小分别为 k_1,k_2 和 k_3,$k_1+k_2+k_3=k$. 我们用 U_j 和 V_j 分别表示所有在 S_j 中的操作中出现的 T_1 和 T_2 中的结点集. 则由结构索引的构造定义可知 U_j 和 V_j 要么其一为空,要么分别对应 I_1 中的结点 u_{Ij} 和 I_2 中的结点 v_{Ij}. 若 U_j 为空,则将 $\Lambda\rightarrow v_{Ij}$ 加入 S_I 中,若 V_j 为空,则将 $u_{Ij}\rightarrow\Lambda$ 加入 S_I 中,否则,将操作 $u_{Ij}\rightarrow v_{Ij}$ 加入 S_I 中. 于是,我们可以得到一个与 S 之对应的 XED 操作系列 S_I. 进一步,我们有

$$\begin{aligned}\gamma(S) &= \sum_{i=1}^{k} S_k \\ &= \sum_{j=1}^{k_1} r(U_j\rightarrow V_j)+\sum_{j=1}^{|k_2|} r(U_j\rightarrow\Lambda)+\sum_{j=1}^{|k_3|} r(\Lambda\rightarrow V_j) \\ &= \sum_{j=1}^{k_1} r(u_{Ij}\rightarrow v_{Ij})+\sum_{j=1}^{|k_2|} r(u_{Ij}\rightarrow\Lambda)+\sum_{j=1}^{|k_3|} r(\Lambda\rightarrow v_{Ij}) \\ &= \gamma(S_I).\end{aligned}$$

证毕.

在引理 2 和引理 3 的基础上,我们进一步证明两棵 XML 树 T_1 和 T_2 间的 XED 距离可以用它们的结构索引 I_1 到 I_2 间的距离来度量.

定理 4.　给定两棵 XML 树 T_1 和 T_2 和它们的结构索引 I_1 到 I_2,任意一个从 T_1 到 T_2 的 XED 映射 M 都有一个从 I_1 到 I_2 的 XED 映射 M_I 与之对应,满足:

(1) $\gamma(M_I)\leq\gamma(M)$;

(2) M_I 是从 I_1 到 I_2 的代价最小的 XED 映射,当且仅当 M 是从 T_1 到 T_2 的代价最小的 XED 映射.

反之,任意一个从 I_1 到 I_2 的 ED 映射 M_I 都有一个从 T_1 到 T_2 的 XED 映射 M 与之对应,且满足上边的条件(1)和(2).

证明. 我们按如下 4 个步骤来证明上边的结论.

(1)根据引理 1,给定一个从 T_1 到 T_2 的 XED 映射 M,都存在一个从 T_1 到 T_2 的 XED 操作系列 S 满足 $\gamma(S)=\gamma(S_I)$. 而由引理 3 可知,存在一个从 I_1 到 I_2 的 XED 操作系列 S_I,满足 $\gamma(S_I)=\gamma(S)$. 再由引理 1,存在一个与 S_I 对应的从 I_1 到 I_2 的 XED 映射 M_I,使得 $\gamma(M_I)\leq\gamma(S_I)$,因此,$\gamma(M_I)\leq\gamma(M)$.

(2) 同理可得,任给定一个从 I_1 到 I_2 的 XED 映射 M_I,都存在一个从 T_1 到 T_2 的 XED 映射 M 满足 $\gamma(M)\leq\gamma(M_I)$.

(3) 我们证明若 M_I 是从 I_1 到 I_2 的代价最小的 XED 映射,则 M 也是从 T_1 到 T_2 的代价最小的 XED 映射. 假设结论不成立,则必存在从 T_1 到 T_2 的 XED 映射 M',满足 $\gamma(M')<\gamma(M)$. 由上边的证明(2),必然存在一个从 I_1 到 I_2 的 XED 映射 M'_I,$M'_I\neq M_I$,且满足 $\gamma(M'_I)\leq\gamma(M')<\gamma(M)$. 再由上边的证明(1),$\gamma(M)\leq\gamma(M_I)$,因此,$\gamma(M'_I)<\gamma(M_I)$,与假设矛盾.

(4) 类似地可以证明,若 M 是从 T_1 到 T_2 的代价最小的 XED 映射,则 M_I 也是从 I_1 到 I_2 的代价最小的 ED 映射.　　证毕.

根据上边的定理,容易得出下边的推论.

推论 1.　给定两棵 XML 树 T_1 和 T_2 和它们的结构索引 I_1 到 I_2,$XED(T_1,T_2)=XED(I_1,I_2)$.

该推论说明两棵 XML 树 T_1 和 T_2 间的 XED 距离可以用它们的结构索引 I_1 到 I_2 间的距离来度量.

例 3.　图 1 中的两棵树 T_1 和 T_2 的结构索引如图 3 中的 I_1 和 I_2 所示. 图 3 中的每个索引结点表示一个等价类,由元素名和属于该等价类的元素实例

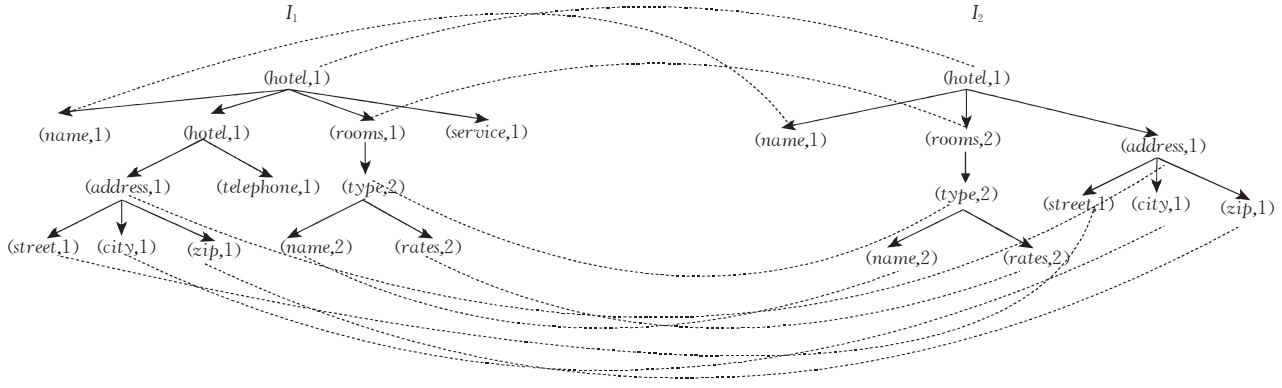

图 3　图 1 中的两棵树的结构索引 I_1,I_2 和它们之间的 XED 映射

的个数(即权重)组成.虚线给出了一个从 I_1 到 I_2 的代价最小的 XED 映射,该映射与例 2 中给出的 T_1 到 T_2 的 XED 映射相对应,二者的代价都为 4.

5 结 论

本文提出了一个适合于定量度量 XML 文档间的相似度的新的编辑距离.利用结点间的相似关系,一个 XML 文档可以表示为一个精简的带权重的结构索引树.两个 XML 文档间的距离可以通过计算它们的索引树间的距离来测定.由于索引树通常远小于原文档的大小,利用索引树可以大大提高判定两个 XML 文档结构相似度的效率.本文将来的工作包括 XML 文档的近似搜索、XML 文档聚类和 XML 文档的结构抽取等.

致谢 加拿大西安大略大学的 Zhang Kai-Zhong 教授对作者的问题给予了热心的解答,并提出了有益的建议,在此表示衷心的感谢.

参 考 文 献

1 XQuery: A query language for XML. W3C Working Draft 15 February 2001, available: http://www.w3.org/TR/xquery/

2 Deutsch A, Fernandez M, Florescu D *et al*. XML-QL: A query language for XML. W3C Note, 1998, available: http://www.w3.org/TR/1998/NOTE-xml-ql-19980819/

3 Robie J, Lapp J, Schach D. XML query language (XQL). W3C Note, 1998, available: http://www.w3.org/TandS/QL/QL98/pp/xql.html

4 Zhang C, Naughton J F, DeWitt D J *et al*. On supporting containment queries in relational database management systems. In: Proceedings of ACM SIGMOD Conference, Santa Barbara, CA, USA, 2001.425～436

5 Florescu D, Kossmann D, Manolescu I. Integrating keyword search into XML query processing. In: Proceedings of the 9th International WWW Conference, Amsterdam, Netherlands, 2000

6 Tai K C. The tree-to-tree correction problem. Journal of the ACM, 1979, 26(3): 422～433

7 Wang J T-L, Zhang K *et al*. Exact and approximate algorithms for unordered tree matching. IEEE Transactions on Systems, Man and Cybernetics, 1994, 24(4): 668～678

8 Zhang K, Shasha D. On the editing distance between unordered labeled trees. Information Processing Letters, 1992, 42(3): 133～139

9 Zhang K. A constrained editing distance between unordered labeled trees. Journal of Algorithmica, 1996, 15(3): 205～222

10 Wang J T-L, Shasha D *et al*. Structural matching and discovery in document databases. Sigmod Record, 1997, 26(2): 560～564

11 Zheng Shi-Hui, Zhou Ao-Ying *et al*. Structure-based approximate searching in XML data. Fudan University: Technical Report TR20010203,2001.

12 Goldman R, Widom J. Summarizing and searching sequential semistructured sources. Stanford University: Technical Report TR20000312, 2000

13 Marian A, Abiteboul S, Cobena G, Mignet L. Change-centric management of versions in an XML warehouse. In: Proceedings of the 27th International Conference on Very Large Data Bases, Roma, Italy,2001.581～590

14 Henzinger M R, Henzinger T A, Kopke P W. Computing simulations on finite and infinite graphs. In: Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, 1995.453～462

15 Tarjan. Three partition refinement algorithms. SIAM Journal on Computing, 1987, 16(6): 973～989

ZHENG Shi-Hui, born in 1974, Ph. D.. His research interests include XML/Web databases, data warehouse, and E-Commerce.

ZHOU Ao-Ying, born in 1965, professor and Ph. D. supervisor. His research interests include database, data mining, and E-Commerce.

ZHANG Long, born in 1976, master. His research interests include database and Web information management.