



(12) 发明专利申请

(10) 申请公布号 CN 103294791 A

(43) 申请公布日 2013. 09. 11

(21) 申请号 201310192029. X

(22) 申请日 2013. 05. 13

(71) 申请人 西安电子科技大学

地址 710071 陕西省西安市太白南路 2 号

(72) 发明人 霍红卫 郭海涛 高培 张懿璞

于强 孙春晓 郭鸿志

(74) 专利代理机构 陕西电子工业专利中心

61205

代理人 田文英 王品华

(51) Int. Cl.

G06F 17/30 (2006. 01)

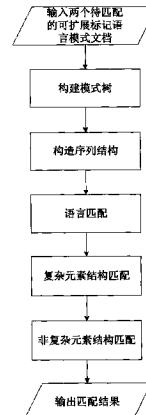
权利要求书5页 说明书10页 附图1页

(54) 发明名称

一种可扩展标记语言模式匹配方法

(57) 摘要

本发明公开一种可扩展标记语言模式匹配方法,用于解决现有技术在模式表示形式、发现复杂匹配、匹配效率等方面的问题,其步骤为:输入可扩展标记语言模式;构建模式树;构造序列结构;对所有元素进行名称、数据类型和基数约束匹配,得出所有元素对的语言相似值;对复杂元素,进行孩子相似性、叶子相似性、兄弟相似性和祖先相似性匹配,获得复杂元素对的结构和整体相似值,过滤找出匹配的复杂元素对;对每个匹配的复杂元素对,找出每个元素对应的原子集,对原子集中的元素,应用非复杂元素结构匹配方法,计算出原子元素的结构和整体相似值,过滤找出匹配的元素对;输出所有匹配元素对。本发明全程自动化,在保证匹配质量的前提下提高了匹配效率。



1. 一种可扩展标记语言模式匹配方法,包括以下具体步骤:

(1) 输入两个待匹配的可扩展标记语言模式文档;

(2) 构建模式树:

将两个待匹配的可扩展标记语言模式文档进行文档对象模型解析,生成两个待匹配的可扩展标记语言模式文件的模式树;

(3) 构造序列结构:

分别对两个模式树进行普吕弗序列构造,获得由编号普吕弗序列和标记普吕弗序列组成的两个加强普吕弗序列;

(4) 语言匹配:

4a) 分别从两个加强普吕弗序列的标记普吕弗序列中任意选取一个元素 s 和元素 t;

4b) 采用名称相似值计算方法,获得元素 s 和元素 t 的名称相似值;

4c) 采用数据类型相似值计算方法,获得元素 s 和元素 t 的数据类型相似值;

4d) 采用基数约束相似值计算方法,获得元素 s 和元素 t 的基数约束相似值;

4e) 将元素 s 和元素 t 的名称相似值、数据类型相似值、基数约束相似值的加权平均数作为元素 s 和元素 t 的语言相似值;

4f) 重复执行步骤 4a) 至步骤 4e),直到得到两个标记普吕弗序列中所有元素两两之间的语言相似值;

(5) 复杂元素结构匹配:

5a) 按照节点在模式树中的后序号从小到大的顺序,分别对两个加强普吕弗序列中的编号普吕弗序列的所有节点进行排序;

5b) 分别从两个排序后的编号普吕弗序列中任意选取一个元素 i 和元素 j;

5c) 采用孩子相似值计算方法,获得元素 i 和元素 j 的孩子相似值;

5d) 采用叶子相似值计算方法,获得元素 i 和元素 j 的叶子相似值;

5e) 采用兄弟相似值计算方法,获得元素 i 和元素 j 的兄弟相似值;

5f) 采用祖先相似值计算方法,获得元素 i 和元素 j 的祖先相似值;

5g) 将元素 i 和元素 j 的孩子相似值、叶子相似值、兄弟相似值、祖先相似值的加权平均数作为元素 i 和元素 j 的结构相似值;

5h) 将元素 i 和元素 j 的结构相似值和步骤 (4) 获得的语言相似值的加权平均数作为元素 i 和元素 j 的整体相似值;

5i) 重复执行步骤 5c) 至步骤 5h),直到得到两个排序后的编号普吕弗序列中所有元素两两之间的整体相似值;

5j) 对两个排序后的编号普吕弗序列中所有元素两两之间的整体相似值,使用阈值法进行过滤,得到所有匹配的复杂节点对,组成匹配的复杂节点对集;

(6) 非复杂元素结构匹配:

6a) 从复杂节点结构匹配所得到的匹配元素对中任取一个元素对,将元素对中的元素分别记为元素 e 和元素 f;

6b) 分别搜索元素 e 和元素 f 所在的加强普吕弗序列,找出元素 e 和元素 f 的所有原子,组成元素 e 和元素 f 的原子集;

6c) 从元素 e 的原子集中,任取一个元素 c,采用非复杂元素结构匹配方法,获得元素 c

与元素 f 的原子集中所有元素的结构相似值；

6d) 判断元素 e 的原子集中是否还有元素,如果有,则执行步骤 6a);否则,认为已得到了元素 e 与元素 f 的原子集中所有元素两两之间的整体相似值,执行步骤 6e);

6e) 重复执行步骤 6a)、步骤 6b)、步骤 6c)、步骤 6d),直到得到所有复杂节点结构匹配所得到的匹配元素对所对应的原子集中所有元素两两之间的整体相似值；

6f) 对所得到的所有元素对的整体相似值,使用阈值法进行过滤,得到匹配的非复杂节点对,组成匹配的非复杂节点对集；

(7) 输出匹配结果：

输出步骤(5)得到的匹配的复杂节点对集和步骤(6)得到的匹配的非复杂节点对集的并集。

2. 根据权利要求 1 所述的一种可扩展标记语言模式匹配方法,其特征在于,步骤(3)中所述的普吕弗序列构造的具体步骤如下：

第 1 步,搜索两个待匹配的可扩展标记语言模式文档的模式树,从中找到具有最小后序遍历顺序号的叶子节点；

第 2 步,将所找到的具有最小后序遍历顺序号的叶子节点,存储于标记普吕弗序列中,同时将该叶子节点的父节点存储于编号普吕弗序列中；

第 3 步,从两个待匹配的可扩展标记语言模式文件的模式树中,删除所找到的具有最小后序遍历顺序号的叶子节点；

第 4 步,判断两个待匹配的可扩展标记语言模式文件的模式树是否为空,如果是,则执行第 1 步;否则,完成了普吕弗序列的构造。

3. 根据权利要求 1 所述的一种可扩展标记语言模式匹配方法,其特征在于,步骤 4b) 中所述的名称相似值计算方法的实现步骤如下：

第 1 步,按照名称令牌化规则,将元素 s 和元素 t 的名称进行分割,得到令牌集 1 和令牌集 2；

第 2 步,从令牌集 1 中任取一个元素 a,采用决策树方法,获得元素 a 与令牌集 2 中所有元素的字符串相似值；

第 3 步,找出第 2 步中所得到的相似值中的最大值,将该最大值作为元素 a 与令牌集 2 的字符串相似值；

第 4 步,判断令牌集 1 中是否还有元素,如果有,则转到第 2 步;否则,认为已得到了令牌集 1 所有元素与令牌集 2 的字符串相似值,执行第 5 步；

第 5 步,将令牌集 1 所有元素与令牌集 2 的字符串相似值累加,得到累加和；

第 6 步,从令牌集 2 中任取一个元素 b,采用决策树方法,计算元素 b 与令牌集 1 中所有元素的字符串相似值；

第 7 步,找出第 6 步中所得到的相似值中的最大值,将该最大值作为元素 b 与令牌集 1 的字符串相似值；

第 8 步,判断令牌集 2 中是否还有元素,如果有,则转到第 6 步;否则,认为已得到了令牌集 2 所有元素与令牌集 1 的字符串相似值,执行第 9 步；

第 9 步,将令牌集 2 所有元素与令牌集 1 的字符串相似值累加,得到累加和；

第 10 步,将第 5 步、第 9 步中所得到的累加和相加,得到总和；

第 11 步, 将总和除以两令牌集中元素的总个数, 所得商即为元素 s 和元素 t 的语言相似值。

4. 根据权利要求 1 所述的一种可扩展标记语言模式匹配方法, 其特征在于, 步骤 4c) 中所述的数据类型相似值计算方法是指, 判断元素 s 和元素 t 的数据类型是否均为内置类型, 如果是, 则从类型相似表中找出元素 s 和元素 t 的数据类型相似值; 否则, 判断元素 s 和元素 t 的数据类型是否均为复杂类型, 如果是, 则元素 s 和元素 t 的数据类型相似值为 1; 否则, 元素 s 和元素 t 的数据类型相似值为 0。

5. 根据权利要求 1 所述的一种可扩展标记语言模式匹配方法, 其特征在于, 步骤 4d) 中所述的基数约束相似值计算方法是指,

第 1 步, 根据元素 s 和元素 t 的基数约束的不同取值, 判断元素 s 和元素 t 的基数约束是否均为基本基数约束值, 如果是, 则查找基数约束相似表, 得出元素 s 和元素 t 的基数约束相似值; 否则, 执行第 2 步至第 4 步, 计算元素 s 和元素 t 的基数约束相似值:

第 2 步, 使用下式计算元素 s 和元素 t 的最小基数约束相似值:

$$u = 1 - \frac{|x - y|}{x + y}$$

其中, u 表示元素 s 和元素 t 的最小基数约束相似值, x 表示元素 s 的最小约束出现次数, y 表示元素 t 的最小约束出现次数;

第 3 步, 使用下式计算元素 s 和元素 t 的最大基数约束相似值:

$$v = 1 - \frac{|m - n|}{m + n}$$

其中, v 表示元素 s 和元素 t 的最大基数约束相似值, m 表示元素 s 的最大约束出现次数, n 表示元素 t 的最大约束出现次数;

第 4 步, 计算元素 s 和元素 t 的最小基数约束相似值和最大基数约束相似值的平均值, 该平均值即为元素 s 和元素 t 的基数约束相似值。

6. 根据权利要求 1 所述的一种可扩展标记语言模式匹配方法, 其特征在于, 步骤 5c) 中所述的孩子相似值计算方法的实现步骤如下:

第 1 步, 利用加强普吕弗序列所包含的父子关系特性, 搜索元素 i 所对应的加强普吕弗序列, 找出元素 i 的所有孩子, 组成元素 i 的孩子集; 搜索元素 j 所对应的加强普吕弗序列, 找出元素 j 的所有孩子, 组成元素 j 的孩子集;

第 2 步, 从元素 i 的孩子集中, 任取一个元素 p, 得到元素 p 与元素 j 的孩子集中所有元素的整体相似值;

第 3 步, 找出第 2 步中所得到的相似值中的最大值, 将该最大值作为元素 p 与元素 j 的孩子集的相似值;

第 4 步, 判断元素 i 的孩子集中是否还有元素, 如果有, 则转到第 2 步; 否则, 认为已得到了元素 i 孩子集中所有元素与元素 j 的孩子集的整体相似值, 执行第 5 步;

第 5 步, 将元素 i 的孩子集中所有元素与元素 j 的孩子集的所有相似值累加, 得到一个累加和;

第 6 步, 将第 5 步所得到的累加和除以两个孩子集所含元素数的最大值, 所得商即为元素 i 和元素 j 的孩子相似值。

7. 根据权利要求 1 所述的一种可扩展标记语言模式匹配方法, 其特征在于, 步骤 5d) 中所述的叶子相似值计算方法的实现步骤如下:

第 1 步, 利用加强普吕弗序列所包含的父子关系和兄弟关系特性, 搜索元素 i 所对应的加强普吕弗序列, 找出元素 i 的所有叶子, 组成元素 i 的叶子集; 搜索元素 j 所对应的加强普吕弗序列, 找出元素 j 的所有叶子, 组成元素 j 的叶子集;

第 2 步, 以元素在模式树中的后序号与元素的叶子集中每个叶子节点在模式树中的后序号的差作为元素的数字向量的分量, 分别构建元素 i 和元素 j 的数字向量;

第 3 步, 使用余弦定理, 计算元素 i 和元素 j 的数字向量的相似值, 该相似值即为元素 i 和元素 j 的叶子相似值。

8. 根据权利要求 1 所述的一种可扩展标记语言模式匹配方法, 其特征在于, 步骤 5e) 中所述的兄弟相似值计算方法的实现步骤如下:

第 1 步, 利用加强普吕弗序列中所包含的兄弟关系特性, 搜索元素 i 所对应的加强普吕弗序列, 找出元素 i 的所有兄弟, 组成元素 i 的兄弟集; 搜索元素 j 所对应的加强普吕弗序列, 找出元素 j 的所有兄弟, 组成元素 j 的兄弟集;

第 2 步, 从元素 i 的兄弟集中, 任取一个元素 q, 得到元素 q 与元素 j 的兄弟集中所有元素的语言相似值;

第 3 步, 找出第 2 步中所得相似值中的最大值, 将该最大值作为元素 q 与元素 j 的兄弟集的相似值;

第 4 步, 判断元素 i 的兄弟集中是否还有元素, 如果有, 则转到第 2 步; 否则, 认为已得到了元素 i 的兄弟集中所有元素与元素 j 的语言相似值, 执行第 5 步;

第 5 步, 将元素 i 的兄弟集中所有元素与元素 j 的兄弟集的所有相似值累加, 得到一个累加和;

第 6 步, 将第 5 步所得到的累加和除以两个兄弟集所含元素数的最大值, 所得商即为元素 i 和元素 j 的兄弟相似值。

9. 根据权利要求 1 所述的一种可扩展标记语言模式匹配方法, 其特征在于, 步骤 5f) 中所述的祖先相似值计算方法的实现步骤如下:

第 1 步, 利用加强普吕弗序列中所包含的父子关系特性, 搜索元素 i 所对应的加强普吕弗序列, 找出元素 i 的所有祖先, 并按照搜索的先后顺序将元素 i 的所有祖先连接起来构成元素 i 的祖先路径; 搜索元素 j 所对应的加强普吕弗序列, 找出元素 j 的所有祖先, 并按照搜索的先后顺序将元素 j 的所有祖先连接起来构成元素 j 的祖先路径;

第 2 步, 将祖先路径看作一个字符串序列, 路径中的每个节点名称看作一个整体, 利用语言匹配方法得到的语言相似值, 计算元素 i 的祖先路径和元素 j 的祖先路径之间的编辑距离;

第 3 步, 将第 2 步中得到的元素 i 的祖先路径和元素 j 的祖先路径之间的编辑距离除以元素 i 的祖先路径长度(所含祖先节点数)和元素 j 的祖先路径长度中的最大值, 得到一个商;

第 4 步, 单位 1 减去第 3 步所得到的商, 即为元素 i 和元素 j 的祖先相似值。

10. 根据权利要求 1 所述的一种可扩展标记语言模式匹配方法, 其特征在于, 步骤 6c) 中所述的非复杂元素结构匹配的实现步骤如下:

第 1 步,从元素 t 的原子集中任取一个元素 d ;

第 2 步,采用权利要求 8 所述的兄弟相似值计算方法,获得元素 c 与元素 d 的兄弟相似值;

第 3 步,采用权利要求 9 所述的祖先相似值计算方法,获得元素 c 与元素 d 的祖先相似值;

第 4 步,将元素 c 与元素 d 的兄弟相似值和祖先相似值的加权平均值,作为元素 c 与元素 d 的结构相似值;

第 5 步,将元素 c 与元素 d 的结构相似值和语言相似值相加,所得到的和作为元素 c 与元素 d 的整体相似值。

## 一种可扩展标记语言模式匹配方法

### 技术领域

[0001] 本发明属于通信技术领域,更进一步涉及数据处理技术领域中的一种可扩展标记语言 (eXtensible Markup Language XML) 模式匹配方法。本发明可根据模式的名称和结构信息,对两个输入可扩展标记语言模式文档自动进行可扩展标记语言模式匹配,找出两个文档中所有相似元素之间的映射,用于确定不同可扩展标记语言数据之间的相似性。

### 背景技术

[0002] 随着 Internet 的发展,可扩展标记语言应运而生并成为了网络中数据表示、数据分析和数据交换的标准。由于可扩展标记语言数据描述的灵活性,可扩展标记语言文档数量和规模的日益增大,如何高效的管理大规模可扩展标记语言数据以及集成大量的可扩展标记语言数据资源变得十分重要。因此,用于识别可扩展标记语言模式之间元素一致性的可扩展标记语言模式匹配技术成为研究热点。

[0003] 可扩展标记语言模式匹配以两个可扩展标记语言模式作为输入,使用不同的相似值计算方法得到两个可扩展标记语言模式之间的一个映射。可扩展标记语言模式匹配在数据共享应用领域发挥着重要作用:在数据集成中,它可用于识别并标记多个模式之间的内部模式关系;在数据仓库中,它能够将一个数据资源映射到仓库模式;在电子商务中,它可以实现不同可扩展标记语言格式之间的消息映射;在语义网络中,它可以用建立不同网站的本体概念之间的语义对应关系;在数据迁移中,它能够将来自多个资源的遗留数据迁移为一个新的数据;在数据转换中,它能够将一个源对象映射为目标对象;在 XML 数据集群中,它可以用确定不同可扩展标记语言数据之间的语义相似性。

[0004] 早期的模式匹配通常是手工完成的,手动指定模式匹配是一个浪费时间、容易出错并且开销很大的过程。当前,大量自动模式匹配算法和匹配系统相继提出,如 LSD(Learning Source Descriptions), Cupid, COMA(COmbination of Matching algorithms), Similarity Flooding, AgreementMaker, ASMOV(Automated Semantic Matching of Ontologies with Verification), OII Harmony 等。现有的大量模式匹配算法和系统虽然实现了模式的半自动或全自动匹配,匹配质量也较高,但可扩展标记语言自动模式匹配中仍然存在许多缺陷。首先,大部分匹配算法仅发现简单匹配(1:1 匹配),发现复杂匹配仅有较少的方法。其次,大部分匹配算法主要考虑模式之间的整体相似性,忽略了独立元素之间的相似度,而可扩展标记语言模式的元素相似性研究能够很好的支持半自动和劳动密集型活动,比如可扩展标记语言模式集成。最后,也是最重要的是大部分匹配系统仅仅关注匹配质量,忽略了匹配效率,使得大规模数据的匹配效率极低。比如元素名称匹配中借助外部词典(WordNet 等)进行语义相似匹配,这虽然提高了名称匹配的准确率,但频繁的查词会大大增加匹配时间。

[0005] 南开大学提出的专利申请“基于扩展邻接矩阵的 XML 文档结构及语义相似性计算方法”(申请号 201010118060.5 申请公布号 CN101799825A) 公开了一种基于扩展邻接矩阵的可扩展标记语言文档结构及语义相似性计算方法。该方法的具体步骤是:第一,输入可扩

展标记语言文档，并对可扩展标记语言文档树进行编码；第二，对于编码后的两个文档，生成模式文档节点列表和数据源文档节点列表；第三，基于所生成的两节点列表，生成模式扩展邻接矩阵和数据源扩展邻接矩阵；第四，使用余弦定理计算两邻接矩阵的距离，得出两个可扩展标记语言文档的相似值。该专利申请存在的不足是：首先，该方法仅在文档层次上度量模式的相似性，而未深入到文档的元素这一更细的粒度上，这就使得该方法不能用于基于可扩展标记语言模式元素间映射的数据处理应用中；其次，该方法仅使用节点标签、节点层次信息、节点编码信息和节点的父节点信息这些有限的信息，作为度量节点的相似性的依据，可能会在相似值计算中产生较大的误差。

[0006] MITRE CORP[US] 提出的专利申请“TOOLS AND METHODS FOR SEMI-AUTOMATIC SCHEMA MATCHING”(申请号 US20060491167 申请公布号 US2008021912A1) 公开了一种半自动的可扩展标记语言模式匹配工具和方法，具体步骤是：第一，输入待匹配的源和目标可扩展标记语言模式；第二，图形化显示源和目标可扩展标记语言模式；第三，询问用户是否希望手工指定某些匹配；如果是，则让用户在所显示的可扩展标记语言模式图上手工指定某些匹配；否则，执行第四步至第七步；第四，对源和目标可扩展标记语言模式，进行语言预处理，并由一组匹配投票器进行打分；第五，由投票合并器对第四步所得到的所有得分进行合并，生成匹配矩阵；第六，加入结构信息进一步调整分值；第七，图形化显示匹配的结果；第八，重复执行第三步至第七步，直到计算出所有的匹配得分。该专利申请存在的不足是：尽管在人工干预下，半自动化的可扩展标记语言模式匹配方法在某种程度上可以提高模式匹配的质量，但是这只能适应于小规模的数据处理，对于较大规模的可扩展标记语言模式文档，手工指定模式匹配是一个单调乏味，浪费时间并且容易出错的过程，因此，这可能会限制该方法所能处理的可扩展标记语言模式数据的规模。

[0007] MICROSOFT CORP[US] 提出的专利申请“METHODS AND SYSTEMS FOR MODEL MATCHING”(申请号 US20010028912 申请公布号 US2003120651A1) 公开了一种模型或模式匹配的方法和系统，具体步骤是：第一，输入待匹配的源和目标可扩展标记语言模式；第二，对输入的两个可扩展标记语言模式进行文档对象模型解析；第三，将生成的文档对象模型转化为通用对象模型；第四，进行根属性匹配和结构匹配；第五，返回匹配结果。该专利申请存在的不足是：结构匹配主要利用叶子节点的相似性，而未全面考虑节点的其它如孩子、兄弟等结构相关信息，这可能会降低结构匹配的质量；其次，在结构匹配中，为改善结构匹配效果，需要重复遍历子树，进行多遍节点相似值的更新，这在一定程度上提高匹配的准确性，但在处理大规模可扩展标记语言模式时，可能会造成很大的系统开销，从而降低了匹配效率。

## 发明内容

[0008] 本发明的目的是针对上述现有技术的不足，提出一种可扩展标记语言模式匹配方法，采用加强普吕弗序列作为可扩展标记语言模式的中间表示，并充分利用语言相关的信息和结构相关的信息，找出两个文档中所有相似元素之间的映射。该方法全程自动化，并在保证匹配质量的前提下提高了匹配效率，解决现有模式匹配在模式表示形式、发现复杂匹配、匹配效率等方面遇到的问题。

[0009] 为了实现上述目的，本发明的具体步骤包括如下：

- [0010] (1) 输入两个待匹配的可扩展标记语言模式文档。
- [0011] (2) 构建模式树：
- [0012] 将两个待匹配的可扩展标记语言模式文档进行文档对象模型解析，生成两个待匹配的可扩展标记语言模式文件的模式树。
- [0013] (3) 构造序列结构：
- [0014] 分别对两个模式树进行普吕弗序列构造，获得由编号普吕弗序列和标记普吕弗序列组成的两个加强普吕弗序列。
- [0015] (4) 语言匹配：
- [0016] 4a) 分别从两个加强普吕弗序列的标记普吕弗序列中任意选取一个元素 s 和元素 t；
- [0017] 4b) 采用名称相似值计算方法，获得元素 s 和元素 t 的名称相似值；
- [0018] 4c) 采用数据类型相似值计算方法，获得元素 s 和元素 t 的数据类型相似值；
- [0019] 4d) 采用基数约束相似值计算方法，获得元素 s 和元素 t 的基数约束相似值；
- [0020] 4e) 将元素 s 和元素 t 的名称相似值、数据类型相似值、基数约束相似值的加权平均数作为元素 s 和元素 t 的语言相似值；
- [0021] 4f) 重复执行步骤 4a) 至步骤 4e)，直到得到两个标记普吕弗序列中所有元素两两之间的语言相似值。
- [0022] (5) 复杂元素结构匹配：
- [0023] 5a) 按照节点在模式树中的后序号从小到大的顺序，分别对两个加强普吕弗序列中的编号普吕弗序列的所有节点进行排序；
- [0024] 5b) 分别从两个排序后的编号普吕弗序列中任意选取一个元素 i 和元素 j；
- [0025] 5c) 采用孩子相似值计算方法，获得元素 i 和元素 j 的孩子相似值；
- [0026] 5d) 采用叶子相似值计算方法，获得元素 i 和元素 j 的叶子相似值；
- [0027] 5e) 采用兄弟相似值计算方法，获得元素 i 和元素 j 的兄弟相似值；
- [0028] 5f) 采用祖先相似值计算方法，获得元素 i 和元素 j 的祖先相似值；
- [0029] 5g) 将元素 i 和元素 j 的孩子相似值、叶子相似值、兄弟相似值、祖先相似值的加权平均数作为元素 i 和元素 j 的结构相似值；
- [0030] 5h) 将元素 i 和元素 j 的结构相似值和步骤 (4) 获得的语言相似值的加权平均数作为元素 i 和元素 j 的整体相似值；
- [0031] 5i) 重复执行步骤 5c) 至步骤 5h)，直到得到两个排序后的编号普吕弗序列中所有元素两两之间的整体相似值；
- [0032] 5j) 对两个排序后的编号普吕弗序列中所有元素两两之间的整体相似值，使用阈值法进行过滤，得到所有匹配的复杂节点对，组成匹配的复杂节点对集。
- [0033] (6) 非复杂元素结构匹配：
- [0034] 6a) 从复杂节点结构匹配所得到的匹配元素对中任取一个元素对，将元素对中的元素分别记为元素 e 和元素 f；
- [0035] 6b) 分别搜索元素 e 和元素 f 所在的加强普吕弗序列，找出元素 e 和元素 f 的所有原子，组成元素 e 和元素 f 的原子集；
- [0036] 6c) 从元素 e 的原子集中，任取一个元素 c，采用非复杂元素结构匹配方法，获得元

素 c 与元素 f 的原子集中所有元素的结构相似值；

[0037] 6d) 判断元素 e 的原子集中是否还有元素,如果有,则执行步骤 6a);否则,认为已得到了元素 e 与元素 f 的原子集中所有元素两两之间的整体相似值,执行步骤 6e);

[0038] 6e) 重复执行步骤 6a)、步骤 6b)、步骤 6c)、步骤 6d),直到得到所有复杂节点结构匹配所得到的匹配元素对所对应的原子集中所有元素两两之间的整体相似值；

[0039] 6f) 对所得到的所有元素对的整体相似值,使用阈值法进行过滤,得到匹配的非复杂节点对,组成匹配的非复杂节点对集。

[0040] (7) 输出匹配结果：

[0041] 输出步骤(5)得到的匹配的复杂节点对集和步骤(6)得到的匹配的非复杂节点对集的并集。

[0042] 本发明与现有技术相比具有以下优点：

[0043] 第一,本发明在可扩展标记语言文档元素的层次上进行可扩展标记语言模式匹配。克服了现有技术中仅在文档层次上度量模式的相似性,而未深入到文档的元素这一更细的粒度上的不足,使得本发明能更好的用于半自动和劳动密集型的任务,如可扩展标记语言模式集成等。

[0044] 第二,本发明在语言匹配中充分利用名字、数据类型和基数约束信息,在结构匹配中充分考虑节点的孩子、兄弟、叶子和祖先节点这些结构相关信息。克服了现有技术中仅根据很少的节点相关的语言和结构信息来计算可扩展标记语言模式相似性的不足,使得本发明提高了可扩展标记语言模式匹配的质量。

[0045] 第三,本发明使用高效的序列结构——加强普吕弗序列表示可扩展标记语言模式,并在语言匹配最关键的部分——名称匹配中采用决策树的原理合并多种字符串匹配算法,以提高匹配效率。此外,在结构匹配中,仅把结构匹配方法应用到匹配复杂元素对的原子元素,而不是计算所有原子元素的结构相似值,这种结构匹配方法易于发现复杂匹配,而又同时能够保证匹配效率。克服了现有技术中只关注匹配质量,而忽略匹配效率的不足,本发明在匹配质量和性能达到一种平衡,使得本发明能适用于更广泛的应用。

## 附图说明

[0046] 图 1 为本发明的流程图。

## 具体实施方式

[0047] 下面结合附图 1 对本发明作进一步的详细描述。

[0048] 步骤 1,输入两个待匹配的可扩展标记语言模式文档。

[0049] 从终端输入两个待匹配的可扩展标记语言模式文档。

[0050] 步骤 2,构建模式树。

[0051] 分别对两个待匹配的可扩展标记语言模式文档进行文档对象模型解析,生成两个待匹配的可扩展标记语言模式文件的模式树。

[0052] 模式树可表示为一个三元组  $T = \{N_T, E_T, Lab_{NT}\}$ ,其中  $N_T = \{n_1, n_2, \dots, n_n\}$  是节点集,节点集中的每个节点唯一的表示模式中的一个对象; $E_T = \{(n_i, n_j) | n_i, n_j \in N_T\}$  是边集, $n_i$  是  $n_j$  的父节点,每条边表示两个节点之间的父子关系; $Lab_{NT}$  是节点标签集,所述标签

是描述节点属性的字符串；模式树中的节点分为两种，原子节点和复杂节点；原子节点是没有出边的叶子节点，表示模式中的简单元素和属性，复杂节点是模式树的内部节点，表示复杂元素。

[0053] 可扩展标记语言模式的结构非常复杂，主要表现为：元素和属性的出现次数往往是一个复杂的正则表达式，基数约束不易确定；共享的全局声明元素、属性和复杂类型的存 在使得可扩展标记语言模式中存在环。因此，在构建模式树之前还应包括简化可扩展标记语言模式。简化步骤依次序包括：建立副本以解决模式中的共享，限定递归次数来解决模式中的无穷递归，用规则简化元素和属性的基数约束。模式树是一种根有向标记树，它反映了可扩展标记语言模式文档的层次结构。

[0054] 步骤 3，构造序列结构。

[0055] 对两个模式树分别采用普吕弗序列 (Prüfer Sequences) 生成方法，生成对应的加强普吕弗序列 (Consolidated Prüfer Sequence CPS)。其中，加强普吕弗序列又由编号普吕弗序列 (Number Prüfer Sequence NPS) 和标已普吕弗序列 (Label Prüfer Sequence LPS) 构成，即  $CPS = \{NPS, LPS\}$ ，它们分别了表示模式树中完全不同的信息，其中，编号普吕弗序列表示模式树的结构信息，标记普吕弗序列表示模式树的语义信息，因此，加强普吕弗序列唯一表示了一棵模式树。加强普吕弗序列有自己独特的优点，包含了节点特性和关系特性。节点特性是指：假设  $n_i$  是模式树中的一个节点， $n_i$  在模式树中的后序号为  $k$ ，则  $n_i$  是原子节点当且仅当  $k$  不属于 NPS； $n_i$  是复杂节点当且仅当  $k$  属于 NPS。关系特性又包括：父子关系特性和兄弟关系特性。父子关系特性是指：设  $NPS_i$  为 NPS 中索引为  $i$  的元素， $LPS_i$  为 LPS 中索引为  $i$  的元素，那么则有  $NPS_i$  节点是  $LPS_i$  节点的父亲节点， $LPS_i$  是  $NPS_i$  的直接孩子节点。兄弟关系特性是指：设  $NPS_i$  和  $NPS_j$  分别为 NPS 中索引为  $i$  和  $j$  的元素， $LPS_i$  和  $LPS_j$  分别为 LPS 中索引为  $i$  和  $j$  的元素，则  $LPS_i$  和  $LPS_j$  是兄弟节点当且仅当  $NPS_i = NPS_j$ 。

[0056] 采用普吕弗序列构造方法为待匹配可扩展标记语言模式文件的模式树构造相应加强普吕弗序列，具体实施过程如下：

[0057] 第 1 步，搜索两个待匹配的可扩展标记语言模式文档的模式树，从中找到具有最小后序遍历顺序号的叶子节点；

[0058] 第 2 步，将所找到的具有最小后序遍历顺序号的叶子节点，存储于标记普吕弗序列中，同时将该叶子节点的父节点存储于编号普吕弗序列中；

[0059] 第 3 步，从两个待匹配的可扩展标记语言模式文件的模式树中，删除所找到的具有最小后序遍历顺序号的叶子节点；

[0060] 第 4 步，判断两个待匹配的可扩展标记语言模式文件的模式树是否为空，如果是，则执行第 1 步；否则，完成了普吕弗序列的构造。

[0061] 步骤 4，语言匹配。

[0062] 语言匹配方法基于节点的名称、节点的数据类型和节点的基数约束的三者的相似性，具体实施过程如下：

[0063] 4a) 分别从两个加强普吕弗序列的标记普吕弗序列中任意选取一个元素  $s$  和元素  $t$ 。

[0064] 4b) 采用名称相似值计算方法，获得元素  $s$  和元素  $t$  的名称相似值。

[0065] 在不考虑数据实例的情况下，节点名称是匹配的一个重要信息。节点名称相似可

以是语义相似,如 People 和 Staff,也可以是结构相似,如 Staff 和 TechnicalStaff。名称的结构相似可用字符串匹配方法,计算两个名称字符串的相似值。语义相似需要借助外部词典,而频繁的查找外部词典会增加匹配的时间,因此考虑到匹配效率和匹配的全自动性,名称匹配仅包括名称的结构相似。名称匹配方法的具体实施过程如下:

[0066] 4b1) 按照名称令牌化规则,将元素 s 和元素 t 的名称进行分割,得到令牌集 1 和令牌集 2。

[0067] 在可扩展标记语言模式中,有些节点的名称较长,有些节点表示的信息相同但为了区分常常带有不同数字序号,有些节点的名称带有特殊的符号。为了使节点名称更好的用于字符串匹配算法,节点名称首先规范为由许多子字符串组成的集合 - 令牌集,每个子字符串叫做令牌。令牌化规则是指:以如“\_”、空格、数字、大写字母等的特殊符号为分隔符,将节点名称分割为令牌,并删除令牌集中的非字母令牌,如数字,特殊符号令牌等。

[0068] 4b2) 从令牌集 1 中任取一个元素 a,采用决策树方法,获得元素 a 与令牌集 2 中所有元素的字符串相似值,本步骤具体实施过程如下:

[0069] 第 1 步,从令牌集 2 中任取一个元素 b。

[0070] 第 2 步,比较元素 a 和元素 b 的字符串值,如果完全相同,则元素 a 和元素 b 的字符串相似值为 1;否则,计算元素 a 和元素 b 的编辑距离相似值。

[0071] 第 3 步,判断编辑距离相似值是否大于等于阈值 0.58,如果是,则元素 a 和元素 b 的字符串相似值为所计算的编辑距离相似值;否则,采用 Jaro-Winkler 算法计算元素 a 和元素 b 的字符串相似值,同时采用 3-gram 算法计算元素 a 和元素 b 的另一个字符串相似值,将两个相似值的加权平均数作为元素 a 和元素 b 的字符串相似值。

[0072] 第 4 步,判断令牌集 2 中是否还有元素,如果有,则转到第 1 步;否则,认为已得到了元素 a 与令牌集 2 中所有元素的字符串相似值,执行步骤 4b3)。

[0073] 4b3) 找出步骤 4b2) 中所得到的相似值中的最大值,将该最大值作为元素 a 与令牌集 2 的字符串相似值。

[0074] 4b4) 判断令牌集 1 中是否还有元素,如果有,则转到步骤 4b2);否则,认为已得到了令牌集 1 所有元素与令牌集 2 的字符串相似值,执行步骤 4b5)。

[0075] 4b5) 将令牌集 1 所有元素与令牌集 2 的字符串相似值累加,得到累加和。

[0076] 4b6) 从令牌集 2 中任取一个元素 b,采用决策树方法,计算元素 b 与令牌集 1 中所有元素的字符串相似值。

[0077] 4b7) 找出步骤 4b6) 中所得到的相似值中的最大值,将该最大值作为元素 b 与令牌集 1 的字符串相似值。

[0078] 4b8) 判断令牌集 2 中是否还有元素,如果有,则转到步骤 4b6);否则,认为已得到了令牌集 2 所有元素与令牌集 1 的字符串相似值,执行步骤 4b9)。

[0079] 4b9) 将令牌集 2 所有元素与令牌集 1 的字符串相似值累加,得到累加和。

[0080] 4b10) 将步骤 4b5)、步骤 4b9) 中所得到的累加和相加,得到总和。

[0081] 4b11) 将总和除以两令牌集中元素的总个数,所得商即为元素 s 和元素 t 的语言相似值。

[0082] 4c) 采用数据类型相似值计算方法,获得元素 s 和元素 t 的数据类型相似值。

[0083] 节点名称虽然是语言匹配的重要信息,但名称匹配得到的映射元素中仍有很多错

误的匹配。为了提高匹配质量,数据类型成为了语言匹配中的又一个可以利用的模式信息。可扩展标记语言模式的数据类型有内置类型和自定义类型两种,内置类型包括 string、int、bool 等,自定义类型包括复杂类型和简单类型,而简单类型归结到底也是内置类型。两个内置类型节点的相似性要大于一个内置类型节点和一个复杂类型节点,两个复杂类型节点的类型相似值由节点的结构决定。数据类型匹配方法的具体实施过程如下:

[0084] 判断元素 s 和元素 t 的数据类型是否均为内置类型,如果是,则查找类型相似表,找出元素 s 和元素 t 的数据类型相似值;否则,判断元素 s 和元素 t 的数据类型是否均为复杂类型,如果是,则元素 s 和元素 t 的数据类型相似值为 1;否则,元素 s 和元素 t 的数据类型相似值为 0。

[0085] 4d) 采用基数约束相似值计算方法,获得元素 s 和元素 t 的基数约束相似值。

[0086] 节点的基数约束信息成是语言匹配中可以利用又一个重要的信息,可扩展标记语言模式用“minOccurs”和“maxOccurs”定义了模式中元素或属性的出现次数。文件类型定义 (Document Type Definition DTD) 中节点的基数表示方法有四种基本基数约束值:“\*”、“?”、“+”和“none”,这四种基本基数约束值对应到可扩展标记语言模式定义 (XML Schema Definition XSD) 中,“none”表示 minOccurs = 1 且 maxOccurs = 1,“?”表示 minOccurs = 0 且 maxOccurs = 1,“\*”表示 minOccurs = 0 且 maxOccurs = unbounded,“+”表示 minOccurs = 1 且 maxOccurs = unbounded。如果两个节点的基数约束都可以表示为这四种基本基数约束值,那么这两个节点的约束相似值只需查找基数约束相似表即可,否则,执行下列步骤,计算元素 s 和元素 t 的基数约束相似值:

[0087] 4d1) 使用下式计算元素 s 和元素 t 的最小基数约束相似值:

$$[0088] u = 1 - \frac{|x - y|}{x + y}$$

[0089] 其中, u 表示元素 s 和元素 t 的最小基数约束相似值, x 表示元素 s 的最小约束出现次数, y 表示元素 t 的最小约束出现次数。

[0090] 4d2) 使用下式计算元素 s 和元素 t 的最大基数约束相似值:

$$[0091] v = 1 - \frac{|m - n|}{m + n}$$

[0092] 其中, v 表示元素 s 和元素 t 的最大基数约束相似值, m 表示元素 s 的最大约束出现次数, n 表示元素 t 的最大约束出现次数。

[0093] 4d3) 计算元素 s 和元素 t 的最小基数约束相似值和最大基数约束相似值的平均值,该平均值即为元素 s 和元素 t 的基数约束相似值。

[0094] 4e) 将元素 s 和元素 t 的名称相似值、数据类型相似值、基数约束相似值的加权平均数作为元素 s 和元素 t 的语言相似值。

[0095] 4f) 重复执行步骤 4a) 至步骤 4e), 直到得到两个标记普吕弗序列中所有元素两两之间的语言相似值。

[0096] 步骤 5, 复杂元素结构匹配。

[0097] 复杂节点的结构匹配方法基于节点的四种结构:孩子、叶子、兄弟和祖先,本步骤具体实施过程如下:

[0098] 5a) 按照节点在模式树中的后序号从小到大的顺序, 分别对两个加强普吕弗序列

中的编号普吕弗序列的所有节点进行排序。

[0099] 5b) 分别从两个排序后的编号普吕弗序列中任意选取一个元素 i 和元素 j。

[0100] 5c) 采用孩子相似值计算方法, 获得元素 i 和元素 j 的孩子相似值。

[0101] 作为元素结构相似值中最主要的部分, 孩子相似值直接反映了元素的基本结构, 本步骤具体实施过程如下:

[0102] 5c1) 利用加强普吕弗序列所包含的父子关系特性, 搜索元素 i 所对应的加强普吕弗序列, 找出元素 i 的所有孩子, 组成元素 i 的孩子集; 搜索元素 j 所对应的加强普吕弗序列, 找出元素 j 的所有孩子, 组成元素 j 的孩子集;

[0103] 5c2) 从元素 i 的孩子集中, 任取一个元素 p, 得到元素 p 与元素 j 的孩子集中所有元素的整体相似值。这里的两元素的整体相似值为两元素的语言相似值和两元素的结构相似值的加权平均数; 因为两元素的结构相似值的是按照节点在模式树中的后序号从小到大的顺序计算的, 所以, 此时已经计算出元素 p 与元素 j 的孩子集中元素的结构相似值, 并且其语言相似值已在步骤(4)中计算出来。

[0104] 5c3) 找出第 2 步中所得到的相似值中的最大值, 将该最大值作为元素 p 与元素 j 的孩子集的相似值。

[0105] 5c4) 判断元素 i 的孩子集中是否还有元素, 如果有, 则转到步骤 5c2); 否则, 认为已得到了元素 i 孩子集中所有元素与元素 j 的孩子集的整体相似值, 执行步骤 5c5)。

[0106] 5c5) 将元素 i 的孩子集中所有元素与元素 j 的孩子集的所有相似值累加, 得到一个累加和。

[0107] 5c6) 将步骤 5c5) 所得到的累加和除以两个孩子集所含元素数的最大值, 所得商即为元素 i 和元素 j 的孩子相似值。

[0108] 5d) 采用叶子相似值计算方法, 获得元素 i 和元素 j 的叶子相似值。

[0109] 本步骤具体实施过程如下:

[0110] 5d1) 利用加强普吕弗序列所包含的父子关系和兄弟关系特性, 搜索元素 i 所对应的加强普吕弗序列, 找出元素 i 的所有叶子, 组成元素 i 的叶子集; 搜索元素 j 所对应的加强普吕弗序列, 找出元素 j 的所有叶子, 组成元素 j 的叶子集。

[0111] 5d2) 以元素在模式树中的后序号与元素的叶子集中每个叶子节点在模式树中的后序号的差作为元素的数字向量的分量, 分别构建元素 i 和元素 j 的数字向量。

[0112] 5d3) 使用余弦定理, 计算元素 i 和元素 j 的数字向量的相似值, 该相似值即为元素 i 和元素 j 的叶子相似值。

[0113] 5e) 采用兄弟相似值计算方法, 获得元素 i 和元素 j 的兄弟相似值。

[0114] 本步骤具体实施过程如下:

[0115] 5e1) 利用加强普吕弗序列中所包含的兄弟关系特性, 搜索元素 i 所对应的加强普吕弗序列, 找出元素 i 的所有兄弟, 组成元素 i 的兄弟集; 搜索元素 j 所对应的加强普吕弗序列, 找出元素 j 的所有兄弟, 组成元素 j 的兄弟集。

[0116] 5e2) 从元素 i 的兄弟集中, 任取一个元素 q, 得到元素 q 与元素 j 的兄弟集中所有元素的语言相似值。

[0117] 5e3) 找出步骤 5e2) 中所得相似值中的最大值, 将该最大值作为元素 q 与元素 j 的兄弟集的相似值。

[0118] 5e4) 判断元素 i 的兄弟集中是否还有元素,如果有,则转到步骤 5e2);否则,认为已得到了元素 i 的兄弟集中所有元素与元素 j 的语言相似值,执行步骤 5e5)。

[0119] 5e5) 将元素 i 的兄弟集中所有元素与元素 j 的兄弟集的所有相似值累加,得到一个累加和。

[0120] 5e6) 将步骤 5e5) 所得到的累加和除以两个兄弟集所含元素数的最大值,所得商即为元素 i 和元素 j 的兄弟相似值。

[0121] 5f) 采用祖先相似值计算方法,获得元素 i 和元素 j 的祖先相似值。

[0122] 本步骤具体实施过程如下:

[0123] 5f1) 利用加强普吕弗序列中所包含的父子关系特性,搜索元素 i 所对应的加强普吕弗序列,找出元素 i 的所有祖先,并按照搜索的先后顺序将元素 i 的所有祖先连接起来构成元素 i 的祖先路径;搜索元素 j 所对应的加强普吕弗序列,找出元素 j 的所有祖先,并按照搜索的先后顺序将元素 j 的所有祖先连接起来构成元素 j 的祖先路径。

[0124] 5f2) 将祖先路径看作一个字符串序列,路径中的每个节点名称看作一个整体,利用语言匹配方法计算每个节点的语言相似值,基于祖先路径中所有节点的语言相似值,计算元素 i 的祖先路径和元素 j 的祖先路径之间的编辑距离;计算编辑距离时,仅考虑节点名称是否语言相似而不要求完全相同。例如,假设两个节点的祖先路径分别为 P0/Orders/shipTo 和 P0/POrders/buyer,其中, P0 与 P0 完全相同, Orders 与 POrders 语言上相似, shipTo 和 buyer 语言上不相似,因此,这两个节点的祖先路径之间的编辑距离为 1。

[0125] 5f3) 将步骤 5f2) 中所得到的元素 i 的祖先路径和元素 j 的祖先路径之间的编辑距离除以元素 i 的祖先路径长度(所含祖先节点数)和元素 j 的祖先路径长度中的最大值,得到一个商。

[0126] 5f4) 单位 1 减去步骤 5f3) 所得到的商,即为元素 i 和元素 j 的祖先相似值。

[0127] 5g) 将元素 i 和元素 j 的孩子相似值、叶子相似值、兄弟相似值、祖先相似值的加权平均数作为元素 i 和元素 j 的结构相似值。

[0128] 5h) 将元素 i 和元素 j 的结构相似值和语言相似值的加权平均数作为元素 i 和元素 j 的整体相似值。

[0129] 5i) 重复执行步骤 5c) 至步骤 5h),直到得到两个排序后的编号普吕弗序列中所有元素两两之间的整体相似值。

[0130] 5j) 对两个排序后的编号普吕弗序列中所有元素两两之间的整体相似值,使用阈值法进行过滤,得到所有匹配的复杂节点对,组成匹配的复杂节点对集。

[0131] 步骤 6,非复杂元素结构匹配。

[0132] 对由复杂节点结构匹配所得到的每个匹配元素对,计算元素对所对应的原子集中所有节点间的结构相似值。这种匹配方除了可以提高匹配效率外,还可以识别出复杂匹配。本步骤具体实施过程如下:

[0133] 6a) 从复杂节点结构匹配所得到的匹配元素对中任取一个元素对,将元素对中的元素分别记为元素 e 和元素 f。

[0134] 6b) 分别搜索元素 e 和元素 f 所在的加强普吕弗序列,找出元素 e 和元素 f 的所有原子,组成元素 e 和元素 f 的原子集。

[0135] 6c) 从元素 e 的原子集中,任取一个元素 c,获得元素 c 与元素 f 的原子集中所有

元素的结构相似值,具体步骤如下:

- [0136] 6c1) 从元素 t 的原子集中任取一个元素 d。
- [0137] 6c2) 采用 5e) 中所述的兄弟相似值计算方法,获得元素 c 与元素 d 的兄弟相似值。
- [0138] 6c3) 采用 5f) 中所述的祖先相似值计算方法,获得元素 c 与元素 d 的祖先相似值。
- [0139] 6c4) 将元素 c 与元素 d 的兄弟相似值和祖先相似值的加权平均值,作为元素 c 与元素 d 的结构相似值。
- [0140] 6c5) 将元素 c 与元素 d 的结构相似值和语言相似值相加,所得到的和作为元素 c 与元素 d 的整体相似值。
- [0141] 6c6) 判断元素 f 的原子集中是否还有元素,如果有,则转到步骤 6c1);否则,认为已得到了元素 c 与元素 f 的原子集中所有元素的整体相似值,执行步骤 6d)。
- [0142] 6d) 判断元素 e 的原子集中是否还有元素,如果有,则执行步骤 6a);否则,认为已得到了元素 e 与元素 f 的原子集中所有元素两两之间的整体相似值,执行步骤 6e)。
- [0143] 6e) 重复执行步骤 6a)、步骤 6b)、步骤 6c)、步骤 6d),直到得到所有复杂节点结构匹配所得到的匹配元素对所对应的原子集中所有元素两两之间的整体相似值。
- [0144] 6f) 对所得到的所有元素对的整体相似值,使用阈值法进行过滤,得到匹配的非复杂节点对,组成匹配的非复杂节点对集。
- [0145] (7) 输出匹配结果:
- [0146] 输出步骤(5)得到的匹配的复杂节点对集和步骤(6)得到的匹配的非复杂节点对集的并集。

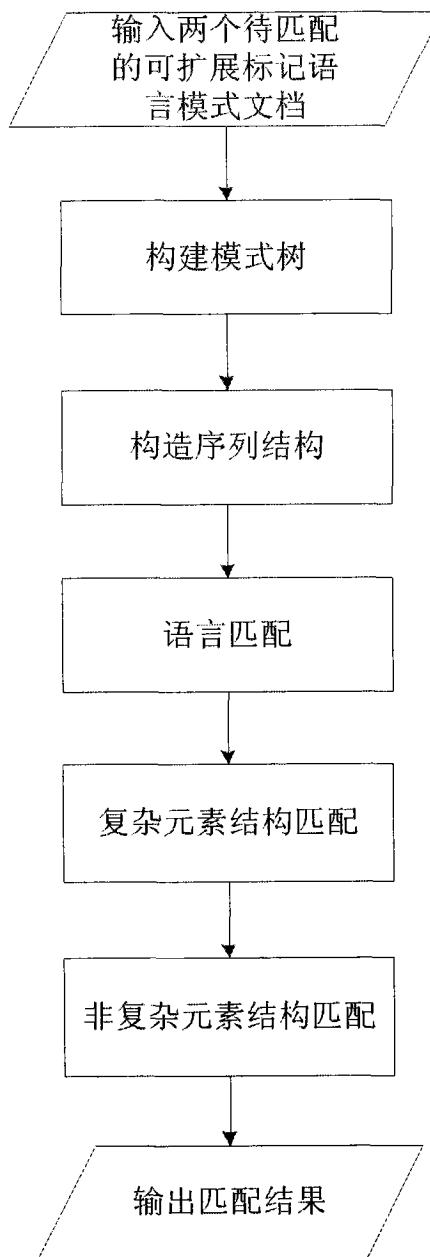


图 1