



一种基于虚拟社交化的 Web 服务发现方法研究 *

潘善亮¹, 茅琴娇², 韩 露¹

(1. 宁波大学信息科学与工程学院 宁波 315211; 2. 西安交通大学电子与信息工程学院 西安 710049)

摘 要:提出一种基于虚拟社交化的 Web 服务发现方法,从相互竞争和相互协作的服务关系中寻找满足用户需求的服务。首先在竞争关系中,通过计算服务之间的文本相似度和结构相似度,选择出一组竞争力强的服务列表,然后根据协作关系,通过计算列表中各服务的交际能力,把交际能力强的服务推荐给用户,实现用户的组合服务流程。通过与其他方法在召回率和准确率方面的比较,验证了本方法的有效性。

关键词:Web 服务器;服务发现;社交化;竞争关系;协作关系

doi: 10.3969/j.issn.1000-0801.2013.12.005

Research on a Web Services Discovery Method Based on Virtual Social Network

Pan Shanliang¹, Mao Qinjiao², Han Lu¹

(1. College of Information Science and Engineering, Ningbo University, Ningbo 315211, China;

2. School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: A Web services discovery method based on virtual social network was proposed, looking for services to meet user requirements from the mutual competition and collaboration service relationship. In the competition, the method selected a group of competitive service list by calculating the text similarity and structural similarity between services, and then according to the cooperation relations, calculated the intercourse skills of the list of all services, recommended the strong communication ability of service to the user, realized the combination of user service process. Comparing the recall rate and accuracy with other methods, the validity of this method is verified.

Key words: Web Services, service discovery, socialization, competition relation, collaboration relation

1 引言

随着 Web 服务的快速发展,企业越来越倾向于把公司的业务流程作为服务发布出去,同时也通过互联网寻找满足特定需求的 Web 服务,使得 Web 服务的发布与组合越来越流行。随着服务数量的日益增长,网络中存在着大量的可用 Web 服务,用户寻找到合适服务的效率和准确率也随之降低。因此有效的 Web 服务发现机制对用户发现合适的服务至关重要^[1]。传统的服务发现机制忽略了服

务之间的关联关系,认为服务之间是相互独立的,使得服务发现的结果不能满足用户的需求。例如,用户计划外出旅行,欲通过网络提供的 Web 服务完成旅行安排,主要包含订票服务和酒店预订服务。在返回的大量服务中,用户根据实际情况选择满足需求的服务。由于 Web 服务之间存在业务流程关系,订票服务与酒店预订服务之间存在相互协作关系(例如,订票服务和特定的酒店之间存在不同的折扣关系),而酒店预订服务之间则存在相互竞争关系^[2]。Web 服务之间的两种关系如图 1 所示。

* 浙江省自然科学基金资助项目(No.LY12F02020),宁波市自然科学基金资助项目(No.2012A610066)

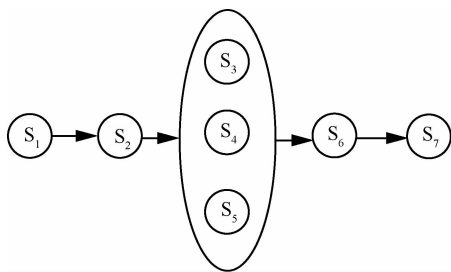


图1 服务之间的竞争与协作关系

图1中, S_3 、 S_4 、 S_5 表示服务之间是相互竞争的关系,箭头方向表示 S_1 、 S_2 、(S_3 、 S_4 、 S_5)、 S_6 、 S_7 服务之间是相互协作的。因为网络中服务数量的增加,相同或相似功能的服务大量存在,某一服务会与其他功能相似的服务形成相互竞争的关系。同样,网络中的Web服务会应服务调用者的要求,与其他服务协作以满足用户关于复杂功能的需求,也就是说,服务之间存在相互协作的关系。

本文把社区网络思想^[3,4]引入Web服务研究中,根据服务之间的相互关系构建一个“社区网络”,然后利用构建的网络实现服务的发现和绑定。主要分析Web服务之间存在相互竞争和相互协作的关系,通过解析这两种关系中Web服务的表现,实现Web服务的发现与选择。具体方法是:竞争关系中的服务发现,计算的是网络中的服务与被请求服务的相似度,与用户需求的服务相似度最高的服务被选出;在协作关系中,计算的是服务的交际能力,与其他服务在相互协作过程中交际能力好的服务被选出,这种交际能力是对服务固有特性的衡量,对于动态环境中的服务发现很重要。因此本文Web服务发现与选择的策略就是把这两种关系中表现好的服务返回给用户^[5,6]。

2 相关工作

目前关于Web服务发现的方法主要分为基于功能性和非功能性的匹配方法,框架如图2所示。

参考文献[7]中Kokas N等人利用基于关键字的匹配方法,除去停用词,取词干以获取Web服务描述文档(Web

services description language,WSDL)中的实义词,再根据这些实义词进行Web服务的语法匹配发现。参考文献[8]中利用关键字匹配中基于向量的方法,构建Web服务文档和查询请求描述文档的向量,然后计算向量之间的相似性,得到服务发现的集合。参考文献[9]中采用了Quality Threshold聚类算法对WSDL文档进行分类,首先从网络上挖掘WSDL文件,得到描述Web服务的content、type、messages、ports和service name这5个部分,然后将这些特征整合起来,按照功能相似度对Web服务进行聚类,不仅可以极大地提高Web服务搜索引擎的准确度,而且能够适应Web服务的动态变化。参考文献[10]提出了基于服务操作匹配的Web服务发现方法,把Web服务描述为一个三元组(ns, ds, p),其中 ns 是该服务的名称, ds 是该服务的文本描述, p 是该服务中操作的集合。因为Web服务发现过程建立在相似匹配的基础上,所以采用基于操作的相似性来实现服务的相似性发现。首先将Web服务内含的操作建模为DOM标签树,然后通过采用满足约束的树编辑距离量化操作与操作之间的相似度,得到服务之间的相似度。参考文献[11]提出了一种基于语义的服务发现方法,通过计算本体树中概念之间的最短路径获得概念间的本体距离,利用获得参数的本体距离,确定服务之间的相似度。

但是,以上关于服务发现的方法均不够全面。参考文献[7]中只考虑基于关键字的功能性匹配,忽略很多语义相似的服务,使得服务发现的结果不够准确。参考文献[9]中采用了对WSDL文档进行分类的方法,大大提高了服务发现的准确度,但是并未考虑服务之间操作的相似度,同样在服务发现的过程中会遗漏很多满足条件的服务。参考文献[10]中通过将服务内含的操作转化为DOM标签树,提高了服务发现的准确率,但是并未综合考虑服务的标签节点和约束节点的计算方法。参考文献[11]中单纯考虑服务发现方法,并未考虑服务之间的调用关系,忽略了服务之间存在着业务上的关联关系。

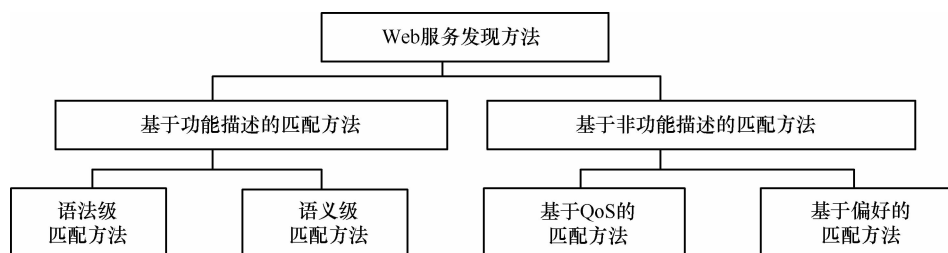


图2 Web服务发现方法的分类



基于以上原因,本文综合考虑了服务之间存在相互竞争和相互协作的关系,在竞争关系中结合用户的需求,计算用户与服务之间文本描述信息的相似度,然后基于 WSDL 的特点,构建 schema 树,计算服务之间的结构相似度,综合考虑文本相似度和结构相似度得到一组服务列表,然后分别计算列表中每个服务的交际能力,选择交际能力强的服务推荐给用户。

3 竞争关系中的服务发现

3.1 问题分析

网络中的 Web 服务是可被公共访问和集成的一个巨大的标准组件库。一个 WSDL 文档与一个 Web 服务相对应,对服务的功能与接口进行描述。Web 服务与一般的 Web 网页包含的大量文本信息不同,只包含非常简短的文本信息,同时含有大量的复杂结构信息。因此 Web 服务的发现不仅要考虑到文本描述信息的相似度,还应该考虑结构的匹配性。

本文在标准的 WSDL 基础上,根据其特点,从 WSDL 文档中解析出关于服务的文本描述信息和服务的结构信息,以实现 Web 服务的发现。同时从用户关于服务的历史调用痕迹中总结出服务所属的兴趣领域,以期返回的服务是用户偏好的服务。

因此,在竞争关系中,Web 服务间相似度的计算综合了服务间的文本相似度与结构相似度。即对于 Web 服务 A 和 B,在竞争的环境下,它们之间的相似度计算如下:

$$ServiceSim(A,B)=\alpha\times ContentSim(A,B)+\beta\times StructureSim(A,B) \quad (1)$$

其中, $\alpha+\beta=1, 0\leq\alpha\leq 1, 0\leq\beta\leq 1$, $ContentSim(A,B)$ 表示两个服务的文本相似度, $StructureSim(A,B)$ 表示服务之间结构的相似度。

3.2 服务的文本相似度

3.2.1 网络中 Web 服务的文本相似度判断

在 Web 服务的相似度发现过程中,网络中的服务应该与被请求的服务是相关的,也就是在基于 WSDL 语言描述的服务中,应该充分挖掘 WSDL 文档中的自然语言信息,并为每个 WSDL 文档构建代表该 Web 服务特征的词汇集合^[12]。

因为 WSDL 文档是结构化的文档,从 WSDL 文档的标签内容中解析出代表服务文本描述的信息。从 WSDL 文档中提取实意词的步骤有以下 5 步。

(1) parsing WSDL: 根据最简单的空格分词法,抽取 <service> 和 <operation> 等标签节点的描述信息,包含服务名称、操作名称和消息参数名称等一系列词汇信息。

(2) tag removal: 去掉得到的词汇信息中的 tag 部分。例如,若在得到的一系列名称信息中包含了 XML、HTML 或者 WSDL 中定义的 tag,如 type、message 等,则删去该信息。因为这些 tag 都是预先定义的,将其删去的工作也比较简单。

(3) word stemming: 把得到的名称信息变为只剩下基词的单词,比如 connect、connected、connecting、connection 的基词都是 connect,本文只取基词,出现次数的多少代表了这个词的重要性。

(4) function word removal: 描述信息中出现的如 of、at、in、without、between 这种辅助词,对服务的发现贡献很小,因此删除此类词。

(5) content word recognition: 一些比较常见的词如 data、web、port 等,这些词在大多数服务中都会出现,因此在服务特征的贡献上没有什么用处,也将其去掉。

经过以上的处理后,就得到文档特征词的集合。

3.2.2 用户历史特性的考虑

考虑用户的历史特性不仅可以挖掘用户潜在的服务偏好,同时可以解决用户描述查询请求不准确的问题,由于专业知识或者使用经验的缺乏,在查询请求中存在不准确的问题。因此本文依据用户调用服务的结果,获取用户历史调用服务的习惯,获得用户的兴趣倾向,实现更准确的服务发现。

本节考虑用户潜在的对服务的偏好信息是通过用户历史调用服务的情况呈现的,也就是用户的这种偏好存在于调用过的服务列表中。通过这个服务列表,挖掘用户的兴趣倾向。可以认为用户的兴趣倾向可以体现在被调用服务的文本描述信息中,即服务名称、操作名称、消息的名称等标签内容中^[13]。因此,用户的历史特性包含在用户调用过的服务的文本描述信息中,这些历史调用信息的获取与前文对网络中服务的文本信息的获取方法类似。

在得到用户的历史特性信息后,结合当前的用户查询请求信息,得到带有用户偏好的查询请求。基于该查询请求,对网络中的服务进行初步的相似度计算,参照传统的向量空间模型实现,计算如下:

$$SR(s,q)=\frac{\sum_{k=1}^n(w_{ik}\times w'_{ik})}{\sqrt{\sum_{k=1}^n(w_{ik})^2}\times\sqrt{\sum_{k=1}^n(w'_{ik})^2}}\tag{2}$$

其中, w_{ik} 代表词汇 k 在服务中的权重, w'_{ik} 代表词汇 k 在请求中的权重。 $SR(s,q)$ 的取值范围为 $(0,1)$, 而且值越大代表用户的请求与当前服务的相似度越大。

3.3 服务的结构相似度

3.3.1 Web 服务的简化

WSDL 是一种基于 XML 的结构化描述语言, 是目前被广泛采用的 Web 服务描述语言^[14]。 WSDL 文档的整体结构如图 3 所示。

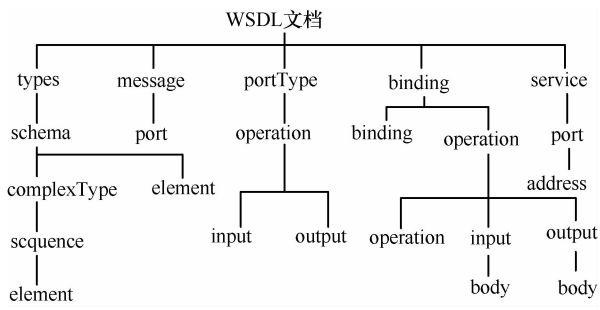


图 3 WSDL 文档结构

WSDL 文档描述服务从两个级别进行。在抽象级别, WSDL 文档通过发送/接收的 message 描述一个 Web 服务, types 定义 Web 服务中的数据类型, 一般使用 XML schema 描述。其中, message 中包含的具体消息内容用 types 元素表示, 而 operation 通过消息交换模式将 message 关联在一起; 在具体级别, WSDL 文档描述的是具体如何与 Web 服务通信, binding 是为 portType 指定传输协议和交换格式, port 把具体网络地址与绑定关联在一起, service 把针对同一 port 的端点组织在一起。

简化后的 WSDL 文档如图 4 所示, 在基于 WSDL 的服务中, Web 服务的功能实体是由执行服务的各个操作完成的, 而对于操作而言, 每个操作又由一组输入参数和一组输出参数组成。输入和输出参数就是用来进行数据交换的消息内容。 Web 服务的操作实例见表 1。

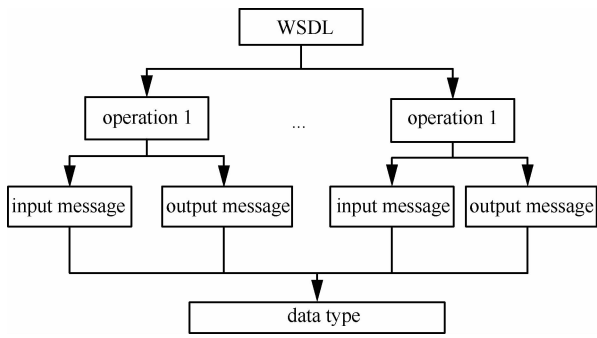


图 4 简化后的 WSDL 文档

3.3.2 Web 服务结构相似度计算

由前文分析可知, Web 服务的实质是实现服务功能的操作列表。 Web 服务之间结构的相似度就是所有服务操作相似度的加权和, 因此得到的是不同操作之间的相似度。 Web 服务操作的相似度就是不同操作之间的输入参数列表和输出参数列表的相似度。 Web 服务操作中的输入参数和输出参数的数据类型通常是基于 XML schema 语法定义的, 包含数据类型和数据结构。数据类型分为简单类型和用户自定义的复杂类型。因此服务操作输入参数中的数据类型与另一服务操作输入参数的数据类型的相似度、服务操作的输出参数的数据类型与另一服务操作的输出参数的数据类型的相似度是 Web 服务结构相似的关键。

本文把操作中的基于 XML schema 描述的数据类型建模为具有标记节点(labeled node)的树状结构。树的节点根据标记(label)的不同, 分为标签节点(tag node)和约束节点(constraint node), 其中标签节点对应于 WSDL 文档中的 element 标签的节点, 约束节点即顺序节点(sequence node)、选择节点(union node)和多选节点(multiplicity node)。顺序节点表示其子节点是按顺序出现的, 选择节点表示该节点的实例只能选择该类型中的一种类型, 多选节点指子节点可以出现的次数范围为 [minOccurs, maxOccurs]。

根据标签树之间的距离计算可以获得基于 XML schema 语法的输入或输出参数的数据类型的相似度。本文

表 1 Web 服务实例

服务名	操作	输入参数	输出参数
W1: GlobalWeathe	GetTemperature	Zip: int	Return: Temperature
W2: CreateOrder	OrderBuilding	UserName: String Requirement: ItemList	ProductList: BuyingOrder
W3: GetLocalTime	LocalTimeByZipCode	Zipcode: int	LocalTimeByZipCodeResult: Time



利用树编辑距离算法计算两棵树之间的相似度,也就是两个服务操作之间的相似度^[15]。

3.3.3 树编辑距离算法

树编辑距离算法是一种常用且有效的计算两树之间相似度的方法。树编辑操作包括删除和修改已存在的节点标签以及插入新的节点标签。两树之间的编辑距离是指两棵树之间的转换所需的最低操作代价^[16]。然而传统的树编辑距离算法无法处理前文中提到的约束节点。应该在保留其功能的情况下,把这些约束节点从树结构中消去。3种约束节点的消去原则如下。

- 顺序节点的分离原则:因为其约束的是子节点出现顺序,因此可以直接消去,但是消去之后对编辑操作的映射代价会有影响。
- 选择节点的合并原则:所有子节点合并为一个节点,将合并的节点的标签名作为新的节点的标签,合并以后对编辑操作的映射代价会有影响。
- 多选节点的删去原则:多选节点也可以直接删去,同样删去之后对编辑操作的映射代价也会有影响。

树编辑距离算法的主要思想是通过节点的删去、插入、替换3种编辑操作后,所需的花费最小。因此计算两棵XML模式树之间的相似度就等价于寻找一个拥有最低代价的映射。而两棵XML模式树标签之间的映射必须满足以下条件。

若 T_x 是一棵树, $T_x[i]$ 是树 T_x 上的第 i 个节点, T_1, T_2 之间的映射是有序的节点对 (i_n, j_n) 的集合, $n=1, 2, \dots$, 它们必须满足以下条件:

- 若 $T_1[i_1]$ 在 $T_1[i_2]$ 的左边, 那么 $T_2[j_1]$ 也在 $T_2[j_2]$ 的左边, 反之亦然;
- 若 $i_1=i_2$, 则 $j_1=j_2$;
- 若 $T_1[i_1]$ 是 $T_1[i_2]$ 的祖先, 那么 $T_2[j_1]$ 也是 $T_2[j_2]$ 的祖先。

两棵树之间的映射实例如图5所示。

由图5可知,如果目标树中出现了源树中没有出现的节点,则是节点的插入;如果源树中的节点在目标树中没有出现则是节点的删去;如果在同一个位置,源树中的节

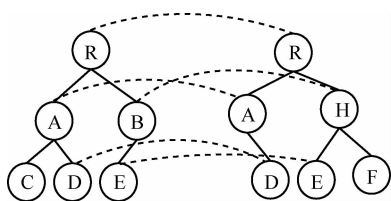


图5 树映射实例

点和目标树中的节点不同则是节点的变换。

3.3.4 编辑操作代价模型

为了寻找编辑操作花费最小的映射,首先应该明确映射中出现的每一步编辑操作的代价,也就是节点的插入、删去以及替换时所需的代价。参考文献[16]中提出的代价模型很简单,认为3种操作的代价均为1,这种算法简单但不准确。参考文献[17]提出对所建的XML模式树中不同层的节点分配一个权重,同时不同层的节点所分配的权重应该不同,而且越靠近根节点的节点获得的权重越大,越往下,权重越小。该方法引入标签节点的权重,对不同编辑操作所需的代价影响不同。

首先考虑节点的插入和删去时两棵树之间相互映射的代价。当进行 schema 树匹配时,节点的插入代价就是节点插入后在目标子树中的权重。而进行节点删除操作,所花费的代价就是节点在源树中的权重。在节点的替换操作中,由于节点处在同一层,权重都一样。但是标签节点由相互连接的单词组成,例如 GetLastTradePrice, 含有丰富的语义信息。因此标签节点在替换操作时,节点之间语义距离的确定是关键。

在服务的发现过程中,服务操作是Web服务器功能属性的简化。因此计算标签节点的替换操作的代价时,考虑这两个标签节点在服务发现时检索的贡献。本文使用正规化 Google 距离(normalized Google distance, NGD)计算两个标签之间的距离。

综合以上分析,本文给出树 T_1 和树 T_2 的编辑操作的代价如下。

- 节点的插入 $Cost(T_1, T_2)$ =节点在目标树中的权重/两棵树所有节点权重之和。
- 节点的删去 $Cost(T_1, T_2)$ =节点在源树中的权重/两棵树所有节点权重之和。
- 节点的替换 $Cost(T_1, T_2)=\{NGD(\text{节点1}, \text{节点2})\}$ 。

其中,关于标签节点的权重定义见参考文献[18]。若树的深度为1,标签节点的权重的值域落在 $[2, 2^1]$ 范围里。因此为了使编辑操作的代价范围为 $[0, 1]$,将节点的编辑操作的代价都除以两棵树的所有节点的权重和。

3.3.5 标签节点之间的语义距离

一般来说,计算词语之间语义距离的方法可以分成两种:一种是基于某种世界知识,如 Ontology 知识,语义相似度的确定依据词汇概念间的结构层次关系确定,但由于构建本体花费代价大,本文不使用此法;另一类就是基于统

计的方法,假设两个词汇出现在相同的上下文中,就认为它们之间具有某种程度的语义相似。

NGD 是指在 Google 搜索中,输入待计算的两个词汇,利用 Google 搜索发现含有这两个词汇的记录数来衡量这两个词汇之间的语义距离^[9]。两个词之间的 Google 距离的计算如下:

$$NGD(a,b) = \frac{\max\{\log f(a), \log f(b)\} - \log f(a,b)}{\log N - \min\{\log f(a), \log f(b)\}} \quad (3)$$

其中, $f(a)$ 指的是返回的 Web 页面包含词汇 a 的页面数, $f(b)$ 指的是返回的页面包含词汇 b 的页面数, $f(a,b)$ 表示的是同时含有词汇 a, b 的 Web 页面数, 而 N 代表 Google 索引到的 Web 页面总数。对于词语 a 和 b , 语义距离为 $NGD(a,b)$ 。NGD 计算的词汇语义距离是对称的, 即词汇 a 到 b 的距离和 b 到 a 的距离一样, 不考虑词汇之间 $a(b)$ 包含 $b(a)$ 的情况。通过这种方式得到的词汇语义距离有以下几个性质。

- 得到的词汇语义距离的取值范围为 $[0, \infty]$ 。另外假设 $a=b$ 或 $a \neq b$, 且 $f(a)=f(b)=f(a,b)>0$, 那么 $NGD(a,b)$ 为 0。代表词汇 a 和 b 含有相同的语义。如果 $f(a)=0$ 或 $f(b)=0$, 那么不论词汇 b 相关的记录数为多少, $f(a,b)=0$, 即 $NGD(a,b)=\infty$ 。
- NGD 为非负数且对任意词汇 a 有 $NGD(a,a)=0$ 。
- 这两个词汇概念的相关度与词汇间的语义距离成反比, 即词汇语义距离值越小, 词汇的相关度越大, 反之亦然。当词汇语义距离为 0 时, 其相似度为 1; 词汇语义距离为 ∞ 时, 其相关度为 0。

Web 服务描述中有很多组合词和合成词, 如果用这些合成词做索引, 搜索查询时很难匹配成功。接下来考虑以下情况。

如果 XML 模式树 T_1 和 T_2 之间存在两对节点之间的映射 (i_1, j_1) 和 (i_2, j_2) , 其中 $T[i_1]=[\text{LocationbyZipCode}]$, $T[j_1]=[\text{ZipCode}]$, $T[i_2]=[\text{date}]$, $T[j_2]=[\text{zone}]$ 。可以看出, 节点对 (i_1, j_1) 因为标签节点存在部分的结构相似, 显而易见, 它们之间的编辑操作应该比节点对 (i_2, j_2) 之间的映射代价要小。而传统的树编辑距离算法没有考虑到标签节点之间的这种差异, 粗略地认为所有标签节点的替换所花费的代价是一样的, 因此是不准确的。

通过前面的分析, 得到了两个单词之间的语义距离。标签节点之间的语义距离应该是构成标签节点的几个单词之间的语义距离的平均。标签节点的划分是依据大写字

母的划分原则, 然后除去停用词。这里标签节点之间语义距离的计算分以下几种情况。

- 如果计算的是类似 ValueName 和 ZipState 这样的标签节点, 构成节点标签的单词都是各自独立的, 无重复。标签节点之间的语义距离就是单词 Value 与 Zip 和 State 的语义距离以及 Name 与 Zip 和 State 的语义距离的和, 除以单词总数的一半。
- 如果计算的两个标签节点的特点与 LocalTimeByZipCode 和 CityStateToZipCode 类似, 除去相同的单词 ZipCode, 计算的是 LocalTime 和 CityState 之间的语义距离。但是它们之间的相似度高, 因此不能忽略相同单词的作用。这样的两个标签之间的距离就是两个不同单词之间的距离再乘以相同单词对在总单词树中的比重, 也就是 $D(a,b)$ (标签节点之间的语义距离) $\times$ 相同单词数 / 总单词数。
- 如果计算的两个标签节点类似 LatestDailyValue 和 DailyValue, 相同的单词就不参与计算, 它们之间的距离统一标记为 1。而总的语义距离就是再乘以相同单词对在总单词树中的比例, 即 $1 \times$ 相同单词数 / 总单词数。

根据标签节点之间的距离 $D(a,b)$, 由式 (4) 计算标签节点之间的语义相似度:

$$SS(a,b) = \frac{\lambda}{D(a,b) + \lambda} \quad (4)$$

其中, λ 是一个可调节的参数, λ 值的范围为 $(0,1)$ 。利用式 (4) 可以使得计算得出的相似度在 $(0,1)$, 因此满足代价要求的范围。

而关于前文约束节点的消去, 对编辑操作产生的影响就是标签节点权重的变化, 本文参考了参考文献 [20] 中的方法。

4 协作关系中的服务发现

4.1 问题分析

由于网络中单个 Web 服务的功能简单, 企业一般都会进行服务的协作以满足应用流程的集成需求。例如在第 1 节的例子中, 旅行安排的服务一般至少包括订票服务和酒店预订服务两种, 通过这两种服务的相互协作完成旅行服务。

协作关系中的 Web 服务, 通过业务逻辑的相互协作, 可



达到业务增值的目的。在返回给用户的服务列表中,用户期望所选择的服务与其他服务有较好的交际能力。这种交际能力不仅体现出业务能力强,而且在完成用户需求的过程中,该服务的等级、权重更高。在服务协作关系分析中,确定一个量化标准来衡量服务的交际能力是关键问题。

4.2 协作关系中服务的交际能力

由前文的分析可知,Web 服务中的数据类型集合和消息集合隐含在服务的操作列表中,所以把 Web 服务简化为一组服务操作的集合 $\{op_1, op_2, \dots, op_n\}$ 。Web 服务提供的功能是通过服务中的操作列表实现的。而 Web 服务的操作又由完成数据交流的输入消息和输出消息组成。执行某一个特定的任务,并将结果以消息的形式返回。服务的操作表示为 $op_i = \{(input_i, output_i, i=1, 2, \dots, n)\}$ 。其中 $input_i$ 是该服务的输入消息的集合, $output_i$ 是该服务的输出消息的集合。WSDL 文档定义的消息交换模式有很多,如 in-out、in-multi-out、out-in 等 10 种交换模式。

Web 服务的交际能力是 Web 服务功能所有操作的交际能力的综合。在 Web 服务的协作关系中,两个 Web 服务的操作可以顺利执行的前提是源操作的输出消息 $output_i$ 与目的操作的输入消息 $input_i$ 相匹配。Web 服务的输出消息与另一个 Web 服务输入消息之间的匹配是两个 Web 服务器连接的纽带,是服务之间数据交流的关键。

但是如果要求一个服务操作的输出集合与另一个服务的输入集合完全相似,如同 Web 服务组合,这样的要求太苛刻。在实际应用中,也就会失去一些本来可以满足用户要求的服务。例如,一个操作的输出集合为 City、ZipCode 和 Temperature,假设网络中没有一个操作的输入集合与之完全匹配,但如果存在一个操作的输入集合为 City、Weather,显然,这样的操作也会满足需求。因此输入与输出不完全相同的服务也可能存在连接关系。尤其在企业的应用中,用户由于专业知识受限或所处的领域不同,在对所需的服务描述不够准确的情况下,不完全匹配可能更符合实际。

因此服务操作之间的连接关系就是源操作的输出集合与目的操作的输入集合之间的相似度。相似度的值就是对存在连接关系的两个服务的连接程度的权重化的衡量。关于相似度的判断,同前文在竞争关系中的计算服务之间的操作的相似度类似,计算的是源操作的输入集合中的数据类型和目的输入集合中的数据类型的相似度。同样计算

的是两个服务操作的数据类型所形成的 schema 树之间的距离。

网络中不同服务之间的连接关系,就像人与人之间的联系。如果一个人的关系网很大,并且相互联系很频繁,或者认识一个交际能力非常好的人,可以通过他来扩大自己的交际圈,本文认为这类人的交际能力很好。在社交网中,服务与服务之间存在连接关系,就像人与人之间的联系。对于一个服务,被调用的次数越多或者与某个交际能力强的服务有交互,就认为这个服务的交际能力强。衡量服务交际能力的思想是基于服务之间的连接关系,这个思想刚好与参考文献[17]中的 PageRank 思想相似。

利用 PageRank 的思想对协作关系中的 Web 服务交际能力进行度量,不仅是基于服务之间的连接关系,而且该思想满足了协作关系中 Web 服务之间交互的 2 个特性。

- 协作关系中的递减属性:假设 Web 服务 1 与 Web 服务 2 连接,而 Web 服务 2 又与 Web 服务 3 连接,由于 Web 服务 3 对 Web 服务 1 的交际能力的评断通过中间服务的连接传递获得,也就不如 Web 服务器 2 这样直接与其交互的服务对其评断来得准确。
- 协作关系中的累加属性:Web 服务与越多的服务形成交互连接,就越增强了该 Web 服务的交际能力。

在协作关系中,基于服务操作之间的连接,对服务的交际能力进行衡量,通过迭代计算,能够相对客观地计算出服务的交际能力。Web 服务操作的交际能力的计算如下:

$$S_{op_i}^t = \frac{(1-\alpha)}{|C_{op_i}|} + \alpha \times \sum_{op_j \in C_{op_i}} Con_{op_j \rightarrow op_i} \times S_{op_j}^{t-1} \quad (5)$$

其中, $S_{op_i}^t$ 表示操作 op_i 的交际能力, t 表示迭代的次数,表示对操作 $Con_{op_j \rightarrow op_i}$ 与操作 op_i 的连接程度的衡量, C_{op_i} 表示与操作 op_i 连接的操作。 α 是一个调节参数, $0 < \alpha < 1$, 表示服务操作的交际能力受其他操作的影响。

服务与服务之间的连接关系与传统网页之间的链接关系不同。服务之间的连接关系不仅表示服务之间有连接,同时连接值的大小会影响交际能力的计算,即服务之间的连接关系是权重化的。

综上所述,网络中的服务存在相互竞争和相互协作的关系。在竞争关系中,解决的是从大量的网络服务中,发现与用户需求描述中的文本和结构相似的服务,得到一组服

务列表。在协作关系中,分别计算列表中服务的交际能力,把网络中交际能力好的服务返回给用户。通过这两种关系实现服务的发现,返回最优的服务给当前用户。

5 实验与分析

因本文提出的是基于 WSDL 文档的功能性 Web 服务发现方法,为了证明该方法的有效性,实验中用到的 Web 服务测试数据均取自真实的 Web 服务库 Webservicelist 和 xMethods。实验过程中共搜集了 280 个来自不同领域的 Web 服务,其中包括 1 478 个 Web 服务操作,作为实验中实现 Web 服务发现和搜索的服务库。鉴于当前 Web 服务发现领域没有统一的衡量标准,本文利用信息检索领域的召回率和准确率作为 Web 服务发现结果的衡量标准。

实验流程如图 6 所示。

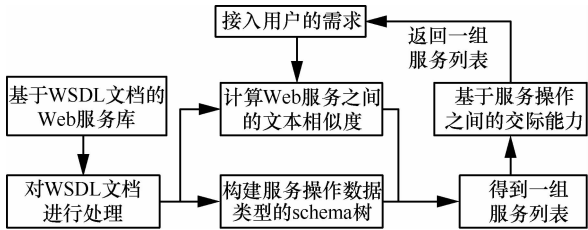


图 6 实验的总体流程

实验中获取的部分 WSDL 文档的信息见表 2,其中包括服务的编号、服务描述文档的 URL、服务的名称、服务提供商以及服务所在的国家。

本文采用通用的 WSDL4J 技术对搜索到的 WSDL 文

档进行解析,WSDL4J 是一个开源的工具包,可以解析出 WSDL 文档所有的信息,如图 7 所示。

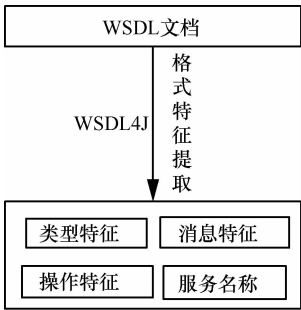


图 7 WSDL 文档解析

实验过程中其他模块的实现如前文所述,其中对特征的提取、标签节点的划分都基于经典算法实现。实验过程中将 30 个 Web 服务、189 个操作作为用户的需求,其他的服务作为服务发现过程中的服务库。实验的召回率和查准率取这 30 次实验的平均。参考文献[9,21]同样是实现基于 WSDL 语言的服务发现。参考文献[9]所使用的 Web 服务发现方法是根据服务功能通过 WSDL 文档对服务进行分类,将该方法记为 CWSD;参考文献[21]采用改进的操作相似性度量方法,记为 MOSM;本文的方法记为 SWSD。实验过程中,在实验的前期对实验的 Web 服务进行分类,依据参考文献[22]中的方法分为 8 类:(1)金融类服务;(2)开发类服务;(3)政府/正常类服务;(4)旅游类服务;(5)电子商务类服务;(6)实体销售类服务;(7)股票/证券类服务;(8)验证类服务。关于服务发现结果的召回率和查准率,实验的结果如图 8、图 9 所示。

表 2 Web 服务信息

WSID	WSDL 文件地址	Web 服务名称	服务提供商	所在国家
0	http://orange-cagnes.dyndns.org/axis/services/wsInvoker?wsdl	WSInvokerService	dyndns.org	法国
1	http://aquest.dyndns.org/CaptchaAudioWS/CaptchaAudio.asmx?WSDL	CaptchaAudio	dyndns.org	日本
2	http://dinoch.dyndns.org:7070/axis1.2/services/EchoService?wsdl	Echo	dyndns.org	美国
3	http://www.smsglobal.com.au/mobileworks/soapserver.php?wsdl	MobileWorks	smsglobal.com.au	澳大利亚
4	http://agiv.be/gis/localService.asmx?WSDL	localService	agiv.be	比利时
5	http://biomoby.org/services/wsdl/ualberta.ca/BacMapGetSpeciesCard	BacMapGetSpeciesCard	ualberta.ca	加拿大
6	http://biomoby.org/services/wsdl/ualberta.ca/BacMapGetGeneCard	BacMapGetGeneCardSer	ualberta.ca	加拿大
7	http://biomoby.org/services/wsdl/ualberta.ca/getColiCardIDs_by_Cofactor	getColiCardIDs_by_Co.	ualberta.ca	加拿大
...	...	...	...	...

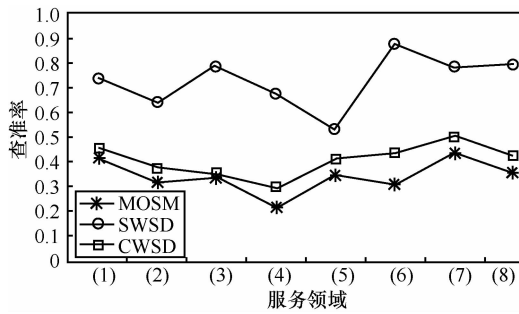


图8 不同服务领域的查准率比较

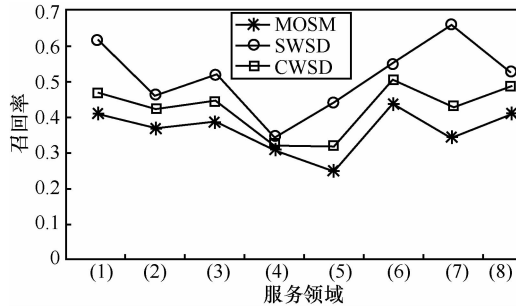


图9 不同服务领域的召回率比较

从图8、图9中可以看出,在查准率和召回率上,本文的方法都有明显提高。可以看出本文的方法充分利用了WSDL的特点,考虑了Web服务竞争关系和协作关系的特性,使返回给用户的服务就是用户倾向的服务,而不是单纯追求返回服务的Top-K集。

6 结束语

为了解决Web服务器发现的难题,本文考虑到网络中某个服务在调用过程中会与其他的服务形成相互竞争或相互协作的关系,实现了一种虚拟社交化的服务发现与选择方法。在相互竞争的关系中,首先通过分析用户历史调用服务情况,同时结合当前用户的需求,计算用户与服务之间文本描述信息的相似度,然后再基于WSDL的特点,构建Web服务操作的schema树,利用改进的树编辑距离算法确定服务之间的结构相似度。其次对于某些用户的复杂需求,需要服务之间的相互协作才能完成。在相互协作的关系中,考虑的是服务的交际能力,这种交际能力反映在服务之间的连接上。最后把交际能力强的服务推荐给当前用户。因此本文利用服务交互调用过程中的这两个特性,从相互竞争和相互协作的关系中寻找满足用户需求的服务。通过将本文的方法与其他方法在召回率和查准率上进行比较,验证了本文方

法的可行性和高效性。

参考文献

- 1 吴朝晖,邓水光,吴健.服务计算与技术.杭州:浙江大学出版社,2009
- 2 Barakat L, Miles S, Luck M. Efficient correlation-aware service selection. Proceedings of IEEE 19th International Conference on Web Service, Honolulu, HI, 2012:1~8
- 3 Maamar Z, Wives L K, Badr Y, *et al.* LinkedWS: a novel web services discovery model based on the metaphor of "social networks". Simulation Modeling Practice and Theory, 2011, 19(1): 121~132
- 4 Maamar Z, Hacid H, Huhns M N. Why Web Services need social networks. Internet Computing, IEEE, 2011, 15(2): 90~94
- 5 Maamar Z, Wives L K, Badr Y, *et al.* Even Web Services can socialize: a new service-oriented social networking model. Proceedings of Intelligent Networking and Collaborative Systems 2009, Madrid, Spain, 2009: 24~30
- 6 Maamar Z, Sheng Q Z, Tata S, *et al.* Towards an approach to sustain Web Services high-availability using communities of Web Services. International Journal of Web Information Systems, 2009, 5(1): 32~55
- 7 Kokash N, Van Den Heuvel W J, D'Andrea V. Leveraging Web Services discovery with customizable hybrid matching. Lecture Notes in Computer Science, 2006, 42(94):522~528
- 8 Platzer C, Dustdar S. A vector space search engine for Web Services. Proceedings of 3rd European Conference on Web Services, Orlando, Florida, USA, 2005:62~71
- 9 Elgazzar K, Hassan A E, Martin P. Clustering WSDL documents to bootstrap the discovery of Web Services. Proceedings of IEEE International Conference on Web Services, Miami, USA, 2010: 147~154
- 10 Hao Y, Zhang Y, Cao J. Web Services discovery and rank: an information retrieval approach. Future Generation Computer Systems, 2010, 26(8):1053~1062
- 11 Rajagopal S, Selvi S, Rajagopalan M R, *et al.* Semantic grid service discovery approach using clustering of service ontologies. Proceedings of IEEE TENCON, Hong Kong, China, 2006:1~4
- 12 W3C Group. Web Services description language (WSDL) 1.1 W3C note 15. <http://www.w3.org/TR/wSDL.html>
- 13 Liu W, Wong W. Web Service clustering using text mining techniques. International Journal of Agent-Oriented Software Engineering, 2009, 3(1):6~26
- 14 Thomas E. SOA 服务设计原则. 郭耀译. 北京: 人民邮电出版社, 2009

- 15 邓水光,尹建伟,李莹等.基于二分图匹配的语义 Web 服务器发现方法.计算机学报,2008,31(8):1364~1375
- 16 Nierman A, Jagadish H V. Evaluating structural similarity in XML documents. Proceedings of 5th Int'l Workshop on the Web and Databases, Madison, Wisconsin, USA, 2002: 61~66
- 17 Xie T, Sha C, Wang X, *et al.* Approximate top-k structural similarity search over XML documents. Proceedings of Frontiers of WWW Research and Development-AP Web 2006, Heidelberg, Berlin, 2006: 319~330
- 18 Brin S, Page L. The anatomy of a large-scale hyper textual Web search engine. Computer Networks and ISDN Systems, 1998, 30(1): 107~117
- 19 张玉芳,艾东梅,黄涛等.结合编辑距离和 Google 距离的语义标注方法.计算机应用研究,2010,27(2):555~562
- 20 Sarwar B, Karypis G, Konstan J, *et al.* Item-based collaborative filtering recommendation algorithms. Proceedings of 10th Int'l WWW Conference, Hong Kong, China, 2001:1~5
- 21 何玲娟,刘连臣,吴澄.一种改进的基于 WSDL 描述的操作相似性度量方法.计算机学报,2008,31(8):1331~1339
- 22 Saha S, Murthy C A, Pal S K. Classification of Web Services using tensor space model and mouth ensemble classifier. Proceedings of the 17th Int'l Symp on Methodologies for Intelligent Systems, Washington DC, USA, 2008:508~513

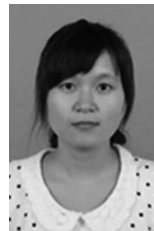
[作者简介]



潘善亮,男,博士,宁波大学副教授,主要研究方向为服务计算、信息检索。



茅琴娇,女,西安交通大学博士研究生,主要研究方向为个性化推荐、用户兴趣建模、统计学习。



韩露,女,宁波大学硕士研究生,主要研究方向为服务计算。

(收稿日期:2013-09-01)

(上接第15页)

- 2 Tormod N, Elin K, Hanne S. Identifying and interpreting market segments using conjoint analysis. Food Quality and Preference, 2001,12(2):133~143
- 3 Kuo R J, Ho L M, Hu C M. Cluster analysis in industrial market segmentation through artificial neural network. Computers & Industrial Engineering, 2002,42(2-4):391~399
- 4 Sineenard P. A mobile information service adapted to vague and situational requirements of individual. Proceedings of the 7th International Conference on Mobile Data Management, Nara, Japan, 2006
- 5 Lai X. Ranking of customer requirements in a competitive environment. Computers & Industrial Engineering, 2008 (54), 202~214
- 6 经有国,但斌,张旭梅等. MC 半结构化客户需求信息表达与处理方法. 管理科学学报, 2011,14(1):78~85
- 7 李桥兴,刘思峰,张旭梅等. 基于区间距和区间侧距的初等关联函数构造. 哈尔滨工业技术学报, 2006, 38(7):1097~1100
- 8 Prasad B. Review of QFD and related deployment techniques. Manufacture Syst, 1998(17):221~234
- 9 Ettlie J E, Johnson M D. Product development benchmarking versus customer focus in applications of quality function deployment. Marketing Lett, 1994(5): 107~116
- 10 陈少威,郭连水,顾子平. 产品需求分析中需求映射方法的研究与实现. 计算机辅助工程, 2002(2):11~16

研究与实现. 计算机辅助工程, 2002(2):11~16

- 11 梁樛,周俊,罗彪. MC 模式下基于顾客需求的产品配置优化分析. 管理科学学报, 2003,6(3):52~56
- 12 Qi Z F, Ruan Y Y. The study of coupling degree of manufacturing cluster. International Conference of Artificial Intelligence and Computational Intelligence (AICI), Sanya, China, 2010:458~462
- 13 唐方成,马骏,席西民. 和谐管理的耦合机制及其复杂性的涌现. 系统工程理论与实践, 2004(11): 68~75

[作者简介]



吴清烈,男,博士,东南大学电子商务系教授,主要研究方向为电子商务、云计算和个性化服务。

郭昱,女,东南大学电子商务系硕士研究生,主要研究方向为云计算和大规模定制。

(收稿日期:2013-05-31)