

基于混合语义相似度的服务匹配方法

熊 芳

(中南大学 湘雅医学院网络中心, 湖南 长沙 410008)

摘 要: 服务匹配已经成为服务计算领域的一大热点, 合理的服务匹配算法可以帮助服务需求者寻找到满足自己需求的服务. 在分析现有的服务算法基础上, 提出基于混合语义相似度服务匹配方法 MSSMA(Mix Semantic Similarity Matching Algorithm). 将比较成熟的语义相似度算法和基于匹配等级的计算方法相结合, 从而提高服务匹配的精度.

关键词: 服务计算; 服务匹配; 语义相似度

中图分类号: TP391

文献标识码: A

文章编号: 1000-7180(2011)08-0083-04

An Approach of Service Matching Method Based on Mixed Semantic Similarity

XIONG Fang

(Network Center of XiangYa Medicinal Colleague, Central South University, Changsha 410008, China)

Abstract: Service matching increasing becomes the one of the hotspots of the service computing. Rational service matching arithmetic can help users find the service they need most. This paper proposed an approach of service matching method based on mixed semantic similarity named Mix Semantic Similarity Matching Algorithm (MSSMA). The method combine the semantic similarity which is comparative mature with computing method based on the matching rank, improving the accurateness of service matching.

Key words: service computing; service matching; semantic similarity

1 引言

服务匹配的计算已成为服务计算领域的一大热点. 文献[1-2]基于词法匹配技术提出服务相似性的判断方法. 文献[3-4]等基于语义推理提出基于目标服务匹配方法. 文献[5-6]等基于 PI 演算和有限状态自动机对服务匹配进行了探讨. 文献[7]通过描述服务抽象模型, 利用语义相似概念对服务进行匹配.

2 研究现状

目前国内外对于基于语义相似度服务匹配的研究如下:

服务匹配选择的依据就是语义相似度, 通过服务需求和服务描述之间的相似度来进行服务的匹配选择.

Paolucci^[8]提出了一个语义匹配算法, 利用描述逻辑和本体语言 OWL 来实现对 Web 服务的匹配, 通过识别服务请求与服务发布之间一对一的输入、输出匹配的匹配程度, 来完成服务和请求的匹配.

文献[9]将服务匹配分做四种情况: 完全匹配、正向匹配、逆向匹配与完全不匹配, 通过对匹配类型的区别来加以权重分析得到匹配结果. 加入类匹配的计算, 就可以很容易地得到服务请求下注册服务之间的匹配程度.

文献[10]中以 Paolucci 的算法为基础, 通过建立对服务的输入输出矩阵, 选取匹配服务. 然而该算法以进程代数和线性逻辑为基础, 缺乏足够的表达服务内部概念间的逻辑语义关系. 文献[11-12]通过计算描述领域本体的概念之间匹配程度, 判断服务

之间的匹配程度,将服务匹配问题转换为本体之间概念匹配程度的计算问题.这种方法通过对服务需求和注册服务之间输入输出的描述文档的语义相似度计算来对服务进行匹配.

文献[13]提出了一种语义相似度的计算方法 MD4-Matching 算法,通过计算概念名称、特征属性以及语义邻居之间的相似度.

文献[14]从语义 Web 服务 OWL-S 描述的功能信息和非功能信息分别进行匹配,然后对两方面的匹配结果进行综合,得出最终结果.该方法对语义 Web 的非功能信息利用自然语言的处理成果,使用向量空间模型对描述文本进行匹配度计算.

文献[15]利用语料库统计的方法,利用词语间的相关性来计算词语的相似度,由于服务匹配问题可以转换为本体概念之间相似度计算问题,简单的通过语义相似度的计算,得出的结果直接影响到了服务匹配的效果.

另外,基于语义相似度的服务匹配计算还包括了对于服务描述本体之间的相似度计算,其中包括基于几何距离的相似度^[16],基于属性的相似度^[17],基于信息容量的相似度^[18]等.

以上方法综合语义相似度的服务匹配,缺少充分的实例匹配特性,导致语义上的不完整,匹配精度不高.下文将给出一种新的服务匹配方法. MD4-Matching 算法流程见图 1.

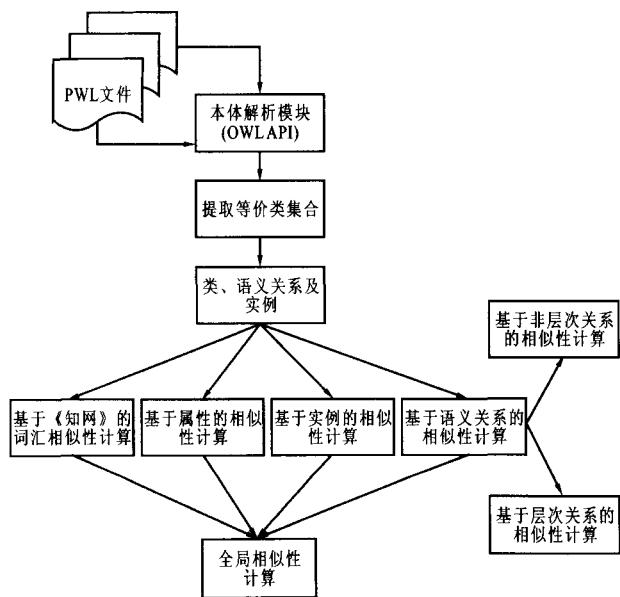


图 1 MD4-Matching 算法流程

3 服务匹配方法

本文服务匹配算法中定义了五个匹配度等级,

它们相当于六个匹配精度不同的过滤器,分别为:精确匹配(EXACT)、嵌入匹配(PLUG-IN)、包含匹配(SUBSUMES)、被包含匹配(SUBSUMED)、前序匹配(PRE-ORDER)和后序匹配(POST-ORDER).前 3 种为基于逻辑的、精度较好的匹配,后 2 种语义匹配是通过交叉匹配前提集和结果集来判定服务之间是否具有顺序关系.

符号“=”、“ $\geq$ ”、“ $\leq$ ”、“ $<$ ”、“ $>$ ”分别表示概念等价、包含、被包含、前序和后序关系.服务请求 R 与服务 S 分别代表的服务需求和注册的某个服务, R 与 S 的各种匹配度定义如下:

定义 1 精确匹配 (EXACT), 称 $R \equiv \text{EXACT}(S)$, 当且仅当 (1) $\forall R_{ci} \in R_{ci}, \forall S_{ci} \in S_{ci}, \exists R_{ci} = S_{ci}$; (2) $\forall R_{co} \in R_{co}, \forall S_{co} \in S_{co}, \exists R_{co} = S_{co}$;

其中 R_{ci} 和 R_{co} 表示请求 R 的输入和输出集, S_{ci} 和 S_{co} 表示请求 R 的输入和输出集. 服务 S 与请求 R 精确匹配, 当且仅当同时满足两个条件, (1) 对服务 S 的任意输入概念, 请求 R 中总存在能与它等价的输入概念; (2) 对请求 R 中的任意输出概念 OUT, S 中总存在相应的输出概念与它等价.

定义 2 嵌入匹配 (PLUG-IN), 称 $R \equiv \text{PLUG_IN}(S)$, 当且仅当 (1) $\forall S_{ci} \in S_{ci}, \exists S_{ci} \in \text{LSC}(R_{ci})$; (2) $\forall R_{co} \in R_{co}, \exists R_{co} \in \text{LSC}(S_{co})$;

嵌入匹配要弱于精确匹配的条件. 嵌入匹配对于输入匹配要求: 对服务 S 中的每个输入概念, 要求在请求 R 的输入概念集中能找到相应的最具体概念子集. 对于输出匹配的要求则是: 对请求 R 的每个输出概念, 要求在服务 S 的输出概念集中能找到相应的最具体概念子集.

下文还将定义其他几个匹配方式, 它们的输入匹配与嵌入匹配要求相同, 但输出匹配要求条件逐渐放松, 匹配的条件更易满足, 返回的结果集也相应增大.

定义 3 包含匹配 (SUBSUMES), 称 $R \equiv \text{SUBSUMES}(S)$, 当且仅当 (1) $\forall S_{ci} \in S_{ci}, \exists S_{ci} \in R_{ci}$; (2) $\forall R_{co} \in R_{co}, \exists R_{co} \in S_{co}$.

包含匹配是进一步弱化的嵌入匹配. 包含对输入匹配的要求: 对服务 S 中的每个输入概念, 要求在请求 R 的输入概念集中能找到相应的概念子集. 对于输出匹配的要求: 对请求 R 的每个输出概念, 要求服务 S 的输出概念集中包含相应的概念子集.

定义 4 被包含匹配 (SUBSUMED) 我们称 $R \equiv \text{SUBSUMED}(S)$, 当且仅当 (1) $\forall R_{ci} \in R_{ci}$,

$\exists R_{\alpha_i} \in S_{\alpha}; (2) \forall S_{\omega_i} \in S_{\omega}, \exists S_{\omega_i} \in R_{\omega}.$

被包含匹配是包含匹配逆向匹配,定义正好与其相反。

定义 5 前序匹配(PRE-ORDER),称 $R \equiv \text{PRE_ORDER}(S)$,当且仅当 $(1) \forall R_{\omega_i} \in R_{\omega}, \exists R_{\omega_i} \in S_{\alpha}; (2) \forall R_{\alpha_i} \in R_{\alpha}, R_{\alpha_i} \notin S_{\omega}.$

这里定义的前序匹配不仅包括了 R 的所有输出集合必须在 S 的输入集合中能够找到相应的概念子集,还规定了匹配之间不能循环,用来防止服务匹配的死锁。

定义 6 后序匹配(POST-ORDER)我们称 $R \equiv \text{POST_ORDER}(S)$,当且仅当 $(1) \forall S_{\omega_i} \in S_{\omega}, \exists S_{\omega_i} \in R_{\alpha}; (2) \forall S_{\alpha_i} \in S_{\alpha}, \exists S_{\alpha_i} \in R_{\omega}.$

后序匹配与前序匹配正好定义相反。

对服务匹配度的定义是用来判断服务与需求之间的匹配程度。下面给出用于基于混合语义相似度的服务匹配算法所需要用到的计算方法。

首先是基于 token 的 Jaccard 两字符串相似度算法。以下给出了该方法计算的一般公式。

$$\text{Sim}_1(S, R) = \frac{\vec{R} \cdot \vec{S}}{\|\vec{R}\|_2 + \|\vec{S}\|_2 - \vec{R} \cdot \vec{S}} \quad (1)$$

在服务描述文档中, S, R 是以一个以空格或回车作为分隔符,由多个单词或词组构成的字符串。

然后,在考虑到服务文档中的本体描述文档中的相似度时,可以利用如下相似度算法:

$$\text{Sim}_2(S, R) = wS_{\text{Name}}(S, R) + uS_{\text{Attribute}}(S, R) + vS_{\text{neighborhoods}}(S, R) + rS_{\text{Inst}}(S, R) \quad (2)$$

其中 $S_{\text{Name}}(S, R)$ 为两个服务描述本体间的名称相似度, $S_{\text{Attribute}}(S, R)$ 为本体间属性相似度, $S_{\text{neighborhoods}}(S, R)$ 为本体间语义环境相似度, $S_{\text{Inst}}(S, R)$ 为本体间实例相似度。

最后,在针对服务匹配的各种类型,我们计算基于服务匹配的计算服务相似度:

$$\text{Sim}_3(S, R) = \begin{cases} 1 & \text{if } R \equiv \text{EXACT}(S); \\ \|\vec{R}\| / \|\vec{S}\| & \text{if } R \equiv \text{PLUG_IN}(S); \\ \|\vec{R} \cap \vec{S}\| / \|\vec{S}\| & \text{if } R \equiv \text{SUBSUMES}(S); \\ \|\vec{R} \cap \vec{S}\| / \|\vec{R}\| & \text{if } R \equiv \text{SUBSUMED}(S); \\ 2 \cdot \|\vec{R}_{\omega} \cap \vec{S}_{\alpha}\| / \|\vec{R}_{\omega} \cup \vec{S}_{\alpha}\| & \text{if } R \equiv \text{PRE_ORDER}(S); \\ 2 \cdot \|\vec{R}_{\alpha} \cap \vec{S}_{\omega}\| / \|\vec{R}_{\alpha} \cup \vec{S}_{\omega}\| & \text{if } R \equiv \text{POST_ORDER}(S); \end{cases} \quad (3)$$

$\|\vec{S}\|$ 表示服务 $\vec{S}$ 的模。

三种关于服务的匹配度算法各有优劣,本文将三种算法作一个线性叠加:

$$\text{Sim}(S, R) = a\text{Sim}_1(S, R) + b\text{Sim}_2(S, R) + c\text{Sim}_3(S, R) \quad (4)$$

其中 a, b, c 的值根据实际情况进行选取。

算法过程如下:

```
Algorithm1: GetServicesimilarity
Input: Web Service Descriptions set
WS = {WS1, WS2, ..., WSn}, Service request: R
Output: servicesimilarity of WS and R
1: begin
2: initiate description WS and R
3: for each web service do
3: for each description document WSdes ∈ WSi
4: do
5: Get token similarity Sim1(WSdes, Rdes)
6: //calculate as equation(1);
7: end for
8: for each description document WSdes ∈ WSi
9: if ontology exists, Get
10: Sim2(WSdes, Rdes)
11: //calculate as equation(2)
12: end for
13: calculate service matching simulation
14: Get: Sim3(WSdes, Rdes)
14: //calculate as equation(3)
15: get Sim(Si, R)
16: //calculate as equation(3)
17: end for
```

通过上述算法,可以计算出服务需求和服务之间的匹配值,从而根据该匹配值选取合适的服务。

4 实验分析

从互联网上,我们可以得到如下 web 服务地址以及相应的服务描述。

(1) <http://12.232.105.153/stockquote/client/form.html>

服务描述: Getting stock quote values from yahoo.

(2) <http://webservices.garsterworks.com/stocks/stocks.asmx>

服务描述: Retrieve the latest stock quotes. Quotes are delayed 20 minutes.

(3) <http://www.webservicex.net/stockquote.asmx?WSDL>

服务描述: Get Stock quote for a company

symbol by using this web service

(4) <http://gotnet.net.innerhost.com/stock/stock.asmx>

服务描述: Stock Quote Service

(5) <http://localhost:8080/soap/servlet/rpcrouter/>

服务描述: Live Quotes From Net-Kraft

(6) <http://services.xmethods.net/soap/urn:xmethods-delayed-quotes.wsdl>

服务描述: Obtain stock quotes.

服务需求描述为用户提出: Stock query 服务请求.

参数 a, b, c 的取值分别为 0.2, 0.5, 0.3, 可以利用上述算法可以得到三种不同算法所得出对于不同服务的相似度, 见表 1.

表 1 相似度比较

	Paolucci	IM-Paolucci	MSSMA
1	0.71	0.78	0.69
2	0.73	0.82	0.71
3	0.71	0.73	0.79
4	0.85	0.86	0.91
5	0.65	0.72	0.63
6	0.81	0.81	0.87

可以看出, Paolucci 和改进的 Paolucci 服务匹配以及本文提出的混合语义相似度算法都能够计算服务与需求的相似度, 但是对于本例而言, 混合语义相似度算法在寻找服务 4 和服务 6 比 Paolucci 算法更具有优势, 能够更加肯定服务需求者的选取.

5 结束语

服务匹配方法是当前服务计算领域研究的热点问题, 高精度的服务匹配算法可以帮助用户查找满意的服务. 本文利用语义相似度算法和传统匹配等级算法相结合的思想, 提出一个合理、有效的服务匹配方法. 在未来的研究工作中, 将重点研究如何在 QoS 监控下的服务匹配计算以及建立更加完善的用户本体描述模型和服务描述模型, 使得服务匹配方法更加有效.

参考文献:

[1] Recommendation. Terms and definitions related to quality of service and network performance including dependency[J]. Theoretical Computer Science, 1994, 2 (6):

115—125.

- [2] Hinterding, Michalewicz, Eiben A. Adaptation in evolutionary computation[C]// 4th International Conference on Evolutionary Computation. Indianapolis, 1997, 5 (13): 65—69.
- [3] 李桢, 杨放春, 苏森. 基于模糊多属性决策理论的语义 web 服务组合算法[J]. 软件学报, 2009, 20 (3): 583—596.
- [4] Wang P, Chao K M, Lo C C. On optimal decision for QoS-aware composite service selection[J]. Expert Systems with Applications, 2009, 12 (3): 114—125.
- [5] Hongxia Tong S Z. A fuzzy multi-attribute decision making algorithm for web services selection based on QoS [C]// IEEE Asia-Pacific Conference on Services Computing. China: Guangzhou, 2006: 205—218.
- [6] Jose Antonio, Parejo P F. Antonio ruiz cort QoS-aware services composition using tabu search and hybrid genetic algorithms[C]// IEEE International Conference on Web Service. China: Beijing, 2008.
- [7] 蒋哲远, 韩江洪, 王钊. 动态的 QoS 感知 Web 服务选择和组合优化模型[J]. 计算机学报, 2009, 32(5): 1018—1021.
- [8] Paolucci T K. Semantic matching of web services capabilities[C]// In Proc of the International Semantic Web Conference. Greek, 2002.
- [9] Gunay A, Yolum P. Structural and semantic similarity metrics for Web service matchmaking [C]// Proc of EC—Web. Germany: Regensburg, 2007: 124—138.
- [10] Baader F. A description logic based approach to reasoning about Web service [C]// Proc of the 14th Int World Wide Web Conf. Japan, Chiba, 2005: 712—725.
- [11] Sirin E. Combining description logic reasoning with AI matching for composition of Web service [D]. Washington: University of Maryland, 2006.
- [12] Boualem Benatallah M S H. On automating web service discovery[J]. The International Journal on Very Large Data Bases, 2005, 7(5): 118—126.
- [13] 黄宏斌. 基于语义关系的战场信息资源聚焦服务方法及关键技术研究 [D]. 湖南, 长沙: 国防科技大学, 2007.
- [14] Jing W, Wu Z. Similarity-based Web Service Matchmaking [C]// IEEE International Conference on Services Computing. USA, Florida, 2005: 259—266.
- [15] Bemstem M K A. Toward high-precision service retrieval[C]// In Proceedings of the First International Semantic Web Conference on the Semantic Web. Greece, Heraklion, 2005: 105—116.

(下转第 89 页)

后删除 v_N 向量. 显然 $v_{\Delta N}$ 也是一个 m -bit 向量, 且至多有 k bits 置为 1.

(4) 基站在该区域广播向量 $v_{\Delta N}$.

(5) 传感器节点根据 $v_{\Delta N}$ 向量更新 v_N 向量. 若 $v'_{\Delta N} = 1$, 则 $v'_N = 0, i \in [0, m-1]$.

这样就完成了一个传感器节点的撤销. 节点撤销后, 也将其对应的 pk_j/sk_j 密钥对销毁.

2.3.2 新节点的加入

假设有新节点 N_{r+1} 要求加入网络, N_{r+1} 首先在基站进行注册, 获得密钥对 (pk_{r+1}, sk_{r+1}) . 基站利用 Bloom Filter 技术, 将 $\langle N_{r+1}, pk_{r+1} \rangle$ 映射到 $\{0 \cdots m-1\}$ 的范围中, 获得更新的 v_N 向量, 利用 Counting Bloom Filter 技术获得更新的 $\overline{v_N}$ 向量. 使用与撤销节点相同的方法, 基站根据更新的 $v_N, \overline{v_N}$ 向量计算得到 $v_{\Delta N}$ 向量, 将 $v_{\Delta N}$ 广播. 节点对向量 v_N 进行更新, 若 $v'_{\Delta N} = 1$, 则 $v'_N = 1, i \in [0, m-1]$. 这样就实现了新节点的加入.

新节点 N_{r+1} 注册完后, 基站计算此节点的单向哈希链 $h(sk_{r+1})$, 并装载在该新节点端. 与 N_{r+1} 进行初次认证时, 基站广播 $h^z(sk_{r+1})$, 依此, 该区域的其他节点知道有新节点的加入. 接下来认证步骤相同.

3 安全性分析与性能分析

3.1 安全性分析

本协议建立在基站是绝对可信的基础上, 恶意节点不可能获得私钥. 在认证过程中, 恶意节点可通过窃听网络流量获得公钥, 但无法计算得到正确的共享密钥, 在用共享密钥验证哈希值时必定失败, 从而取消连接.

本协议的安全性依赖于 Diffie-Hellman 算法的求离散对数的困难性、哈希函数的安全性及节点私钥的保密性. 安全假设建立在只有真正的节点知道私钥的基础上, 所以, 一旦节点的私钥泄露, 协议将失去作用.

3.2 性能分析

在双向认证和密钥协商过程中, 节点的主要计算是一次基于 ECC 的点积计算、一次对称加解密计算和哈希计算. 文献[2]中, 节点的主要计算是一次点积计算、一次 ECDSA 签名和 ECDSA 解签名. 因此, 本协议具有计算量更小、无需携带数字证书的优点. 另一方面, 节点只需存储基站产生的过滤器数据、哈希链值和节点的密钥对.

4 结束语

本文提出了一种适用于无线传感器网络的访问控制协议, 此协议可以实现节点间的双向认证和密钥协商, 当有新节点加入时, 此协议同样发挥作用. 另外, 本协议还便于实现节点的动态管理.

参考文献:

- [1] Zhou Y, Fang Y, Zhang Y. Securing wireless sensor networks: a survey[J]. IEEE Communications Surveys, 2008(10):6-28.
- [2] 郑耿忠, 刘秋梅. 基于模糊控制的移动无线传感器网络部署优化[J]. 微电子与计算机, 2011, 28(2):20-22.
- [3] Zhou Y, Zhang Y, Fang Y. Access control in wireless sensor networks[J]. Security issues in Sensor and Ad Hoc Networks, 2007(5):3-13.
- [4] Ren K, Lou W, Zhang Y. Multi-user broadcast authentication in wireless sensor networks[C]// 4th Annual IEEE Communications Society Conference. San Diego, CA, United States, 2007:223-232.
- [5] Mitzenmacher M. Compressed bloom filters[J]. IEEE/ACM Transactions on Networks, 2002(10):613-620.

作者简介:

谢招彝 男, (1974-), 硕士, 讲师. 研究方向为网络软件与智能信息处理.

郑旋 女, (1984-), 硕士, 助教. 研究方向为计算机网络与安全.

(上接第 86 页)

- [16] Paolucci M K. Semantic matching of Web service capabilities [C]// The First International Semantic Web Conference. Greece, Heraklion, 2002.
- [17] Bansal S M. Matchmaking of web services based on the DAML-S Service Model [C]// 2nd AAMAS. Australia: Victoria, 2003.
- [18] Agarwal S A. Automatic matchinaking of web services

[C]// The 15th International Conference on world wide Web. China: Beijing, 2006.

作者简介:

熊芳 女, (1979-), 工程师. 研究方向为信息服务、语义关联度.