

文章编号: 1000 5013(2010) 05- 0530- 04

多层次语义网的改进 Web 服务检索技术

辛明海, 潘孝铭

(华侨大学 计算机科学与技术学院, 福建 泉州 362021)

摘要: 利用语义 Web 服务中的标准 OWL-S, 提出多层次语义 Web 服务检索技术. 与目前关键字或分类的查询方法相比, 多层次语义 Web 服务检索技术具有更多的判断依据, 并增加 Web 服务在搜寻时的准确率及可用性. 模型包括服务名称分析、概念描述及输出 输出参数相似性分析、服务质量分析 3 个阶段. 实验结果表明, 与传统的基于关键字的检索方式相比, 采用多层次语义扩展检索的查准率平均值上升 0. 25, 查全率平均值上升 0. 24, 但执行效率降低 3. 2%.

关键词: 语义扩展检索; Web 服务; 多层次; 相似度

中图分类号: TP 391. 3

文献标识码: A

随着 Web 服务的迅速发展, 在 Web 服务储存库^[1]中可获取的 Web 服务数量也随之快速增加. 因此, 在服务检索时单纯以关键字搜寻想要的服务, 会对服务的自动组合造成很大的困难. Web 服务架构面临着安全、组合、语义等一些难以解决的问题^[2]. 结合语义网的语义 Web 服务是使用计算机能解释的语言, 描述 Web 服务的能力及内容, 并改进现在的工作质量, 包括服务发现、监视、回复等. 语义 Web 服务技术抽象的架构和所使用的协议, 能够解决 5 种语义 Web 服务代理人的需求: 包括动态服务发现、服务协调、服务规范和管理、社群支持服务、服务质量^[3]. 服务请求者如何才能准确、有效地找到想要的服务, 已经成为重要的研究课题. 用户在 Web 服务注册中心搜寻服务, 其运作机制一般有两种方式, 一种是以 Web 界面的方式, 另一种是以离线网络统一描述、发现和集成协议(UDDI)的方式. 目前, 研究语义 Web 服务的发掘、搜寻或匹配方法, 都是以 OWL(Web Ontology Language)与 Service 结合的 OWL-S 作为标准文件^[4-8]. 本文提出一种多层次语义 Web 服务搜寻机制, 以增加 Web 服务在搜寻时的准确率及可用性.

1 语义网 Web 服务搜索模型的设计

改进文[9] 提出的多层次语义的改进网络匹配方式, 将 Web 服务搜寻机制分为 4 个步骤. 即比对服务名称、服务概念描述内容的相似性分析、输出与输入参数的相似性分析和服务质量分析. 服务需求者的需求先利用 OWL-S 编辑器转换成 OWL-S 文件档, 或将服务储存库中待匹配的 WSDL 文件, 利用套件 CODE^[10]换成 OWL-S 文件, 其流程如图 1 所示.

(1) 比对服务名称. 比对服务名称及分析概念内容描述的相似性是第 1 阶段, 它可以先筛选掉一些不相关的服务, 减少后续的匹配的文件, 减少花费时间; 匹配后再对服务质量分析, 可作为选择的参考依据.

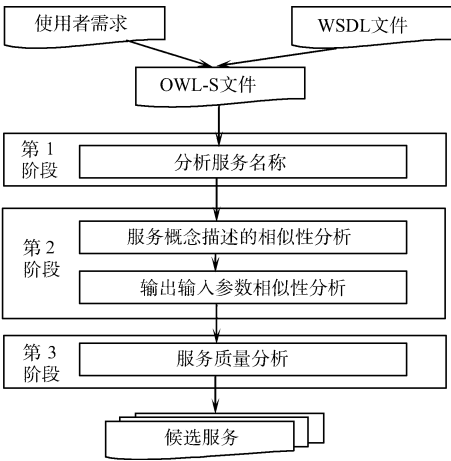


图 1 服务搜寻流程图

Fig. 1 Web services search flow diagram

收稿日期: 2010 01 11

通信作者: 辛明海(1976), 男, 讲师, 主要从事软件工程、模式识别和人工智能的研究. E-mail: minghai_xin@ seu. edu. cn.

基金项目: 福建省自然科学基金资助项目(A0810013); 华侨大学科研基金资助项目(04BS313, 06HZR29); 泉州市科技计划项目(2009- 0032)

为服务名称的命名时, 经常不是单一字词, 如有“GetLocalTime”, “LookupZipcode”这类的复合名词. 因此, 分析前需分解, 将有意义的字词用大写开头及底线“_”相连, 拆解完后, 再继续下一步.

使用者输入关键字的搜寻方式, 可利用近似字符串比对演算法, 将服务储存库中的服务名称与使用者的输入做比对. 近似字符串比对, 可以容忍如下 3 种类型的错误: (1) 置换型错误, 如 computer 变为 cimputer; (2) 遗漏型错误, 如 computer 变为 coputer; (3) 重复型错误, 如 computer 变为 computer.

因此, 使用近似定串比对演算法为编辑距离, 对于 3 种错误类型各视为相差一个距离. 此计算方式会因单词长短的不同而有所误差, 故 3 种情况的总发生率不能大于 0.2. 利用字串相近的方式可增加搜寻弹性, 避免因输入时的小错误而搜寻不到服务.

知识本体的层次架构, 如图 2 所示. 从图 2 可看出, 在名称相似度计算部分, 对于知识本体而言, 阶层越低, 相似性越高. 因此, 对于层级越低的节点, 计算相似度时, 将其权重值提高. 即

$$S(A, B) = d(P(A, B)) \times \frac{1}{d(A, B) + 1} \tag{3}$$

其中: $S(A, B)$ 为两节点的相似度; $d(P(A, B))$ 为 A, B 两节点的共同父节点轿车到根部的距离; $d(A, B)$ 表示 A, B 间的最短路径, 加 1 的目的在避免 A, B 是相同节点时分母为 0.

当分析服务名称完成后, 将服务储存库中的服务分成两类. 较具有相关性名称的服务放入第 2 阶段的流程, 以分析其概念描述及输出参数之相似性; 如果使用者仅以关键字查询方式搜寻服务, 则可至第 3 阶段做服务质量分析.

(2) 服务概念描述的相似性分析. 它是第 2 阶段的第 1 个步骤, 利用文[6]提出的向量空间模型, 依其步骤计算各服务之概念相似性, 其值为 Simprofile.

(3) 输出与输入参数相似性分析. 它是第 2 阶段的第 2 个步骤. 在参数相似性分析部分, 可利用设定字词的权重 $w_i(d)$ 的计算式, 交叉计算需求与储存库中服务的输入与输出参数的相似性. 有

$$w_i(d) = f_i(d) \times \log \left(\frac{N}{d_i} \right).$$

其中: $w_i(d)$ 为字词 i 在文件 d 中的加权值; $f_i(d)$ 为文件 d 中每个字词 i 出现频率; N 为文件总数; d_i 为包含有字词 i 的文件数量.

设需求的输入参数以 R_{IN} 表示、输出参数以 R_{OUT} 表示, 储存库中服务的输入参数以 A_{IN} 表示、输出参数以 A_{OUT} 表示, 分别计算其相似度, 有

$$\begin{aligned} S_{IN-OUT} &= S(R_{OUT}, A_{IN}), \\ S_{OUT} &= S(R_{OUT}, A_{OUT}), \\ S_{IN} &= S(R_{IN}, A_{IN}). \end{aligned}$$

若 S_{IN-OUT} 计算结果相似, 则先将其剔除, 表示两服务工作内容正好相反; 然后, 计算参数的相似度, 有

$$S_p = yS_{IN} + xS_{ON}, \quad x + y = 1.$$

第 2 阶段完成后, 会产生服务集合 S_2 . 集合中, 服务相似度 S_s 的计算方式为

$$S_s = wS_t + zS_p, \quad w + z = 1.$$

(4) 服务质量分析. 前两阶段的相似性分析过程可以视为完整的搜寻步骤. 加入服务质量(QoS)分析^[11]的目的是, 候选服务在做选择的参考排序, 或是当首选服务执行失败时, 可立即找到替代服务.

在服务质量分析项目中, 参考一般关于 Web 服务质量的 QoS 指标, 分别定义如下 7 个 QoS 指标.

- (1) 回应时间. 服务完成所需时间.

(2) 承载量. 服务可同时接受的请求数量.

(3) 可用性. 表示服务是否已准备就绪能直接使用.

(4) 正确性. 表示一段时间内的错误率.

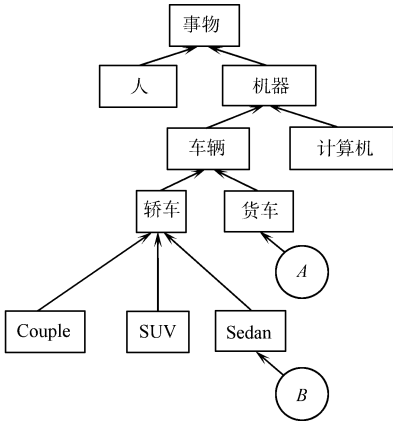


图 2 知识本体的层次架构

Fig. 2 Hierarchical structure of ontology

- (5) 正确性. 表示一段时间内的错误率.
- (6) 成本. 服务所需的价钱.
- (7) 安全性. 表示定义服务是否提供机密性、完整性、可验证性的机制.

服务请求者可以弹性选择其所关注的项目, 并设定其允许极值, 比如服务回应时间在多少时间内等. 最后, 所列出的服务集合即为候选服务, 使用者可从中选取适合的服务. 列出清单的目的在于, 服务请求可依据此清单弹性选择所要的服务, 也可用于 Web 服务的自动化组合.

2 实验结果与分析

采用 Web 服务储存库“XMethod”(<http://www.xmethods.net>), “Web Service List”(<http://www.webservicelist.com/>), 取得旅游相关的 WSDL 文件, 包含订票、天气、购物、汇率等服务, 以及一些不相关的服务文件, 如密钥管理、档案处理等.

首先, 使用 CODE^[10] 将 WSDL 格式的文件转为 OWL-S 格式, 作为实验模拟服务储存库中的服务.

然后, 在搜寻流程第二阶段使用的 w, x, y, z 等 4 个参数. 将使用不同的权重比例测试搜寻结果的准确率, 从而选出最适合计算的公式. 准确率的判断方式以查全率(η_R) 及查准率(η_P) 参照^[12]. 其中: 查全率为候选服务清单中相关服务的数量与服务储存库中所有相关服务数量的比值; 查准率为候选服务清单中相关服务的数量与候选服务清单中的服务数量的比值.

在第 3 阶段的服务质量分析部分, 特别是在目前 OWL-S 标准格式中, 并没有服务质量的相关标签. 因此, 在编辑 OWL-S 文件时, 需再增加 QoS 指标标签.

采用关键字检索与 Web 服务检索的效果对比, 如表 1 所示. 从表 1 可知, 3 种搜索引擎采用关键字检索的查准率平均值为 0.30, 而采用多层次语义扩展检索的查准率平均值为 0.55, 提高了 0.25. Google 搜索引擎的关键字检索的查准率为 0.34, 表明其对中文的分词并不理想, 经多层语义扩展后, 其查准率上升到 0.58, 检索性能显著提升.

表 1 检索方式的效果对比
Tab. 1 Results comparison of retrieval methods

检索方式	Bing 搜索引擎		Yahoo 搜索引擎		Google 搜索引擎	
	η_P	η_R	η_P	η_R	η_P	η_R
关键字检索	0.41	0.21	0.18	0.20	0.34	0.13
语义检索	0.62	0.43	0.44	0.31	0.58	0.39

从表 1 的查全率数值可知, 通过语义扩展可以检索到关键字并不匹配但确为用户所需的服务, 甚至是组合服务, 起查全率平均值上升 0.24, 相比于关键字匹配方式有显著的提高.

采用多层次语义扩展检索做服务搜寻时, 需要执行更多的步骤, 因此系统的执行效率会降低. 实验结果表明, 相比于传统的基于关键字的检索方式, 使用多层次语义扩展检索的执行效率要降低 3.2%.

3 结论

分为 3 个阶段的多层次语义 Web 服务搜寻机制, 相比较于目前关键字或分类的查询方法, 多了更多的判断依据. 第 1 阶段的服务名称分析, 可视为前置的筛选流程, 可以大量减少后续阶段所需匹配的文件数量, 因此能够节省不少时间. 并且, 该阶段的搜寻方式, 相对于只用关键字搜寻的服务请求者而言, 仍然比传统搜寻方式更能找到所需要的服务.

语义相似性比对更切合服务请求者的需求, 增加搜寻到所需要服务的机会; 而增加服务质量的分析, 除了让服务请求者可以多一项参考依据外, 也避免搜寻到不能真正执行的服务.

后继的研究重点是, 考虑此搜寻机制在服务自动化组合时产生的问题, 以及服务质量指标的正确性等问题. 同时, 进一步提高执行效率, 将该搜寻机制扩展应用于如文[13]的软构件检索.

参考文献:

[1] SABOU M, PAN J. Towards semantically enhanced web service repositories[J]. Web Semantics: Science, Services

- and Agents on the World Wide Web, 2007, 5(2) : 142-150
- [2] WANG Hong-bing, HUANG J Z, QU Yu-zhong, et al. Web services: Problems and future directions[J]. Web Semantics: Science, Services and Agents on the World Wide Web, 2004, 1(3) : 309-320.
- [3] BURSTEIN M, BUSSLER C, ZAREMBA M, et al. A semantic Web services architecture[J]. IEEE Internet Computing, 2005, 9(5) : 72-81.
- [4] 陈锦源. 基于语义的 Web 服务发现机制研究[J]. 计算机与现代化, 2007(7) : 34-37.
- [5] MILLER G A. WordNet: A lexical database for English[J]. Communications of the ACM, 1995, 38(11) : 39-41.
- [6] ELHAM P, JAFAR H, SOHEIL Y H. Semantic composability measure for semantic web services[C] // First Asia International Conference on Modelling & Simulation. Phuket: [s. n.], 2007: 88- 93.
- [7] PAOLUCCI M, KAWAMURA T, PAYNE T R, et al. Semantic matching of web services capabilities[C] // Proceedings of the 1st International Semantic Web Conference. Sardinia: [s. n.], 2002: 333-347.
- [8] SU Xiao-meng, GULLA J A. Semantic enrichment for ontology mapping[C] // Proceeding of the 9th International Conference on Applications of Natural Language to Information System. Heidelberg: Springer-Verla, 2004: 217-228.
- [9] 仲梅, 宋顺林. 一种语义 Web 服务的多层次匹配方法[J]. 计算机应用, 2007, 27(1) : 199-201, 204.
- [10] SRINIVASAN N, PAOLUCCI M, SYCARA K. Semantic Web service discover in the OWL-S IDE[C] // Proceedings of the 39th Annual Hawaii International Conference on System Sciences. Washington D C: IEEE Computer Society, 2006: 109.
- [11] XIN Dong, ALON H, MADAHAVAN J, et al. Similarity search for web services[C] // Proceedings of the Thirtieth International Conference on Very Large Data Bases. Toronto: VLDB Endowment, 2004: 372-383.
- [12] SALTON G, MCGILL M J. Introduction to modern information retrieval[M]. New York: McGraw Hill Book Co, 1983.
- [13] 余金山. 利用 XML, Tamino 和 CORBA 的软构件管理与检索技术[J]. 华侨大学学报: 自然科学版, 2008, 29(4) : 518-522.

Improving of Web Services Retrieval Technique by Using Multi-Layer Semantic Web

XIN Ming-hai, PAN Xiao-ming

(College of Computer Science and Technology, Huaqiao University, Quanzhou 362021, China)

Abstract: Multi-level semantic web service retrieval technique was proposed by using the usual standard OWL-S in the semantic web service. Compared with the method simply using keywords or classified mechanism, multi-level semantic web service retrieval technique provides more judgment and improves the precision and usability when searching for Web Services. The model includes 3 phase: service name analysis, concept describing and input-output parameter similar analysis, service quality analysis. Experimental results show that compared with traditional search way based on keywords, using multi-level semantic expansion retrieval method leads the precision ratio average to rise by 0. 25, recall level average to rise by 0. 24, but carry out efficiency to reduce by 3. 2%.

Keywords: semantic expanded retrieval; Web service; multi level; similarity

(责任编辑: 黄仲一 英文审校: 吴逢铁)