基于 WordNet本体和 PLSA的语义 Web服务发现

邢少敏 周伯生 陈天影

(北京航空航天大学 计算机学院, 北京 100191)

摘 要: 提出了一种基于 WordNet本体标注和概率潜在语义分析 (PLSA, Probabilistic Latent Semantic Analysis)的语义 Web服务发现方法 OntoPLSA. 首先使用 WordNet本体标注 Web服务的操作名、参数以及用户请求,以经过标注后的输出参数集合为词汇集,服务描述文档集合为文档集,组成词汇-文档矩阵,以该矩阵为输入,使用 PLSA方法对服务集进行分类,并将用户请求带入 PLSA模型,确定其所属的类;然后在类中以标注后的输出参数为键,含有这个输出的服务的列表为键值,建立一个映射表,查找与用户请求的输出相似的映射表键,进而找出对应的键值,即服务列表;最后根据 QoS(Quality of Service)和用户请求中的输入参数确定满足条件的服务结果集合. 在 415个 Web服务组成的数据集上的测试结果表明,性能较其他方法有优势,召回率和 *R*准确率也得到了改善.

关 键 词: 本体标注; 概率潜在语义分析; Web服务; 索引; 分类; 语义; 服务发现

中图分类号: TP 311

文献标识码: A **文 章 编 号**: 1001-5965(2009)11-1325-05

Semantic Web service discovery based on WordNet ontology and PLSA

Xing Shaomin Zhou Bosheng Chen Tianying

(School of Computer Science and Technology, Beijing University of Aeronautics and Astronautics, Beijing 100191, China)

Abstract: A semantic Web service discovery based on WordNet ontology and PLSA (Probabilistic Latent Semantic Analysis) was proposed. Firstly, the operation name, input parameters, output parameters and the query request were annotated by WordNet ontology. Then the annotated output parameters were taken as term set, and web services as document set to construct a vocabulary-document matrix. After that the query request was taken as service and was injected into the PLSA model to determine its category. Secondly, in this category, a mapping table was established. The output parameters of services were taken as map key, and services list which contained the output parameter was taken as the key value for the map. Each output parameter of query request could match the map keys which were similar to themselves. According to the matched map key, the corresponding list of services would be identified, and then the service set that was compatible with QoS requirements also would be identified. Finally, the final service result set would be acquired according to the query request input parameters. Testing data set including 415 web services was used to make an experiment for this method. Results show that not only the performance is better than other methods but also *R*-precision and recall rate have been improved.

Key words: ontology annotation; probabilistic latent semantic analysis; Web service; indexing; classification; semantics; service discovery

现有的 Web服务发现技术大多基于关键字匹配,如 UDDI[1]等. 关键字匹配方法只能做语法层面的处理,发现精度不高. Woogle[2]使用聚类方法在操作级度量服务相似性,改善了关键字方法

收稿日期: 2008-10-31

基金项目: 北京市科技计划课题资助项目 (y0105001040521); 国家科技部科技型中小企业技术创新基金资助项目 (08c26221100070)

作者简介: 邢少敏 (1980-),男,陕西宝鸡人,博士生, xingshaomin@cse.buaa.edu.cn.

的检索效果和性能. 关键字匹配的另一种替换思路是使用一些技术挖掘隐藏在词汇中的隐式语义,比如概率潜在语义分析 (PLSA, Probabilistic Latent Semantic Analysis)[3],它使用概率方法计算文本潜在语义,降低运算复杂度,提高了发现的准确率和召回率. 但是这种方法并没有利用语义技术,只是单纯从 Web服务的描述文件中提取潜在语义,并且仅仅将 Web服务描述文件当作普通文本来对待,没有考虑 Web服务自身的特征,存在很大的优化改进空间.

针对 Web服务发现所面临的这些问题,提出了一种基于 WordNet本体标注和 PLSA的语义 Web服务发现方法 OntoPLSA.

1 基本思路

基于 OntoPLSA的语义服务发现方法的基本思路如图 1所示. 首先要构建服务检索系统. 在服务检索系统中,首先将服务库中的服务描述文件的相关属性用 WordNet[4]本体库中的本体进行标注,给服务赋予语义概念,便于进行语义级别的匹配和比较,然后构建输出参数—服务矩阵,将每一个服务用一个经过标注后的输出参数向量表示,以该矩阵为输入,建立 PLSA模型,将服务库中的服务分成不同类别,接下来在每一个类别中以 WordNet本体标注后的服务输出参数为键,含有该输出参数的服务列表为键值建立映射索引表,表中服务按照 QoS属性排序,这样服务检索系统就建立了. 当一个服务请求到来时,首先对请求进行 WordNet本体标注,然后基于检索系统的 PLSA模型确定请求所属分类,确定了分类后,在类别中匹配标注后的请求输出参数和映射索引表,匹配成功后再根据 QoS属性确定候选的服务集合,最后根据请求输入参数过滤候选服务集合,得到最终服务结果集合.

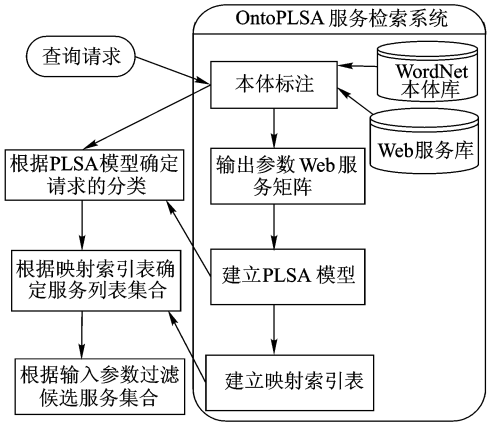

图 1 基本思路图

一些基本概念定义如下:

定义 1 服务 (WS). 服务表示实际 Web服务的一个操作的抽象表示. 可表示为五元组 $WS = \{Name, Des, In, Out, QoS\}$,其中,Name是服务中操作名称; Des是服务描述文本信息; $In = \{In_1, In_2, \cdots, In_m\}$表示输入参数集合; $Out = \{Out_1, Out_2, \cdots, Out_t\}$表示输出参数集合; $QoS = \{P, A, S, C\}$. P (Performance)表示服务的性能属性,也就是响应速度; A (Availability)表示服务可用性,即能够保证可用 (在线)的概率; S (Stability)表示稳定性,即在使用服务时,服务不发生异常的概率; C (Cost)表示使用服务的价格.

定义 2 用户查询请求 (QR). 用户查询请求是对用户发出查询服务请求的抽象描述. 可以表示为四元组 $QR = \{Name_{key}, I, O, QoS\}$. 其中: $Name_{key}$是用户提供的查询操作名; $I = \{I_1, I_2, \cdots, I_h\}$是用户可以提供的输入的集合; $O = \{O_1, O_2, \cdots, O_g\}$是用户需要获得的输出的集合. $QoS = \{P, A, S, C\}$是用户要求的服务要达到的 Qos属性,具体含义同定义 1.

2 Web服务集合的语义标注

2.1 WordNet本体标注

为了进行语义比较,首先要进行 WordNet本体标注,即把服务中的某些词选择语义相近的本体进行标识,表示这些词汇和本体之间语义相近,对于每一个需要标注的词汇,在 WordNet中选择该词汇的 $n(n \geq 1)$级上位词对服务和用户查询请求的操作名称,输入参数和输出参数进行标注.

根据命名习惯,一个操作名或者参数往往不是一个正确的英文词汇,而只是组合的标识符,我们按照下面几个方法来处理: ①单个英文单词:直接用上位词标注; ②英文词组,如 Stock Quote:每个单词独立标注,标注后再组成一个本体词组; ③词组连写,如 StockQuote, Stock_Quote, URL2IP, XML to URL:先用各种分隔符 (下划线,短线),分割词 (2, &, to, from, and)和大小写将词组分割成单词,再对每个单词标注. 忽略分隔符,分割词; ④简写词,如 SMS:不作标注,直接使用简写词; ⑤动作前缀 GetWeather, AddWeather:用大小写分隔成单词,动词不标注,标注名词,再组成语义词组; ⑥找不到上位词的单词:保留原词.

2.2 相似度计算

进行了 WordNet本体标注后,服务属性就可以和请求属性进行匹配,匹配的依据就是各自标注的本体之间的语义相似度,这样请求和服务的

相似度就转化为本体之间的语义相似度. 本体之间的语义相似度是基于语义的匹配问题中相似度计算的基础. 有很多种方法可以使用,这些方法在特定领域被证明是可行的,文献[5]提出了本体之间的相似度的计算公式:

$$S(C_1, C_2) = \begin{cases} e^{-\alpha l} \dfrac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} & C_1 \neq C_2 \\ 1 & C_1 = C_2 \end{cases} \quad (1)$$

其中, l表示 2个概念间的最短路径的长度; h表示 2个概念在树中最近的相同父辈概念在树中的高度; $\alpha, \beta \geq 0$是调节因子,据文献[5]中的实验,当 $\alpha = 0.2, \beta = 0.6$时效果最佳. 可以看出, 2个概念的相似度随 l递减,随 h递增. 该方法通用性较好,容易实现. 因此,本文直接采用该方法计算本体之间的相似度.

3 基于 PLSA的语义发现方法

本文将 PLSA概率潜在语义方法和映射索引表结合起来做服务发现,首先利用潜在语义分析技术进行服务分类,然后利用映射索引表进行服务匹配和查询,提高了服务发现的效率和性能. PLSA基于 aspect model[3]模型,在文本分析中,它使用潜在类别变量 ($Z = \{z_1, z_2, \cdots, z_f\}$)将文本关键字和文本间接的联系起来,并且假设当潜在概念指定时,词和文本是条件独立的,如式(2)~式(3):

$$P(d, w) = P(d) P(w \mid d) \quad (2)$$

$$P(w \mid d) = \sum_{f=1}^{k} P(z_f \mid d) P(w \mid z_f) \quad (3)$$

由式(2)~式(3)又可以得出

$$P(d, w) = \sum P(w \mid z) P(z) P(d \mid z) \quad (4)$$

式(4)中, d表示文档, w表示关键字. PLSA使用概率及统计模型来完成文本分类. 它使用 EM (Expectation Maximization)算法来迭代计算概率分布,在每一个迭代中 E-step中的每一项都可以直接由上一次 M-step中求得的数值来推算. 接下来使用 EM来求出 $P(z)$, $P(w \mid z)$及 $P(d \mid z)$. 对数似然函数为

$$L = \sum \sum n(d_j, w_i) \log P(d_j, w_i)$$

E-step:

$$P(z \mid d, w) = (P(z) P(d \mid z) P(w \mid z)) / \left(\sum P(z_t) P(d_j \mid z_t) P(w_i \mid z_t) \right)$$

M-step:

$$P(w \mid z) = \sum n(d_j, w) P(z \mid d_j, w) / \sum \sum n(d_j, w_i) P(z \mid d_j, w_i)$$

$$P(d \mid z) = \sum n(d, w_i) P(z \mid d, w_i) / \sum \sum n(d_j, w_i) P(z \mid d_j, w_i)$$

$$P(z) = \sum \sum n(d_j, w_i) P(z \mid d_j, w_i) / R$$

$$R = \sum \sum n(d_j, w_i)$$

经过适当的迭代后,其概率会收敛到一定的值. 最后可以算出不同类别在每篇文档中所占的比重:

$$P(z \mid d) = P(d \mid z) P(z) / \sum P(d \mid z_t) \quad (5)$$

想要搜索文章时,可以利用这些概率分布来整理出与查询请求 q相似度最高的文章. 先使用 VSM (Vector Space Model)[6]的方法:将 q视为一篇文章并且建立一个向量,向量的分量是各个词汇出现在该 q里的次数. 接下来将该向量视为测试数据集的一部分并用 EM来训练出 $P(q \mid z)$并计算 $P(z \mid q)$. 由于原本测试数据集的数据规模相当大,因此可以假设这个 q的内容并不会影响到 $P(w \mid z)$, $P(d \mid z)$以及 $P(z)$. 在这样的假设下就可以很快的算出 $P(z \mid q)$. 因为 z有多个,所以使得 $P(z \mid q)$最大的那个 z就是 q所属的分类,知道了 q的分类,就可以使用很多方法在分类中计算 q和其他文章的相似程度,并且依照相似程度输出最后的结果.

3.1 输出参数 Web服务矩阵

经过语义标注后的服务集合组织成一个词汇—文档矩阵,矩阵的列是服务文档,行是所有服务的语义标注后的输出参数,而不是从服务文档中提取所有的词汇,这是因为由于服务查询和普通文本查询的区别. 服务查询并不需要像普通文本查询那样对文本中的每一个关键词进行索引,而只需要对输出索引就可以了,因为用户查询服务最关注的是输出,其他不相关的词汇,比如名称,描述等如果用来做索引,会干扰查询质量,降低准确率. 矩阵中的具体值,仍然使用 TF-IDF[6]方法进行计算. 即 $w_{ij} = t_{ij} \times \log(n / n_i)$. 其中 $t_{ij} = n_{ij} / |a_i|$,表示词频,即词汇 j在服务 i中出现的频率,而 n_i表示包含词汇 j的服务的个数. 表 1是一个示例图.

表 1　输出参数-服务文档矩阵

输出参数	WS_1	WS_2	WS_3	WS_4	WS_5
Out_1	0	0	1.1	0	1.2
Out_2	2.1	0	2.3	2.4	0
Out_3	0	5.1	2.1	0	1.3
Out_4	6.1	0	0	2.0	0

有了这个矩阵以后,每一个服务都可以用一

个向量表示 $WS_i = (Out_1, Out_2, \cdots, Out_k)$,这个向量就代表了服务本身.

3.2 PLSA方法分类

有了服务向量以后,将这些服务向量进行分类,根据式(5),在给定服务文档条件下,计算概率 $p(z|d_{new})$,使得这个概率值最大的那个 z 就是该文档的分类.按照这种方法就可以把文档分为 k 个类别.

$$P(z|d_{new}) = \frac{P(d_{new}|z)P(z)}{P(d_{new})} = \frac{P(d_{new}|z)P(z)}{\sum_{z_j \in Z} P(d_{new}|z)} \quad (6)$$

3.3 输出语义索引

为了提高检索效率和性能,在利用 PLSA 建立为服务建立分类后,对每个类中的服务以输出为 key建立索引表,有几个类就有几个索引表.索引表结构如图 2所示,把服务集合中所有服务的每一个输出(标注后)作为 key,将和这个输出有关的服务列表作为 key值,服务列表按照 QoS属性值由大到小排序,即在图 2中,$Qos(WS_{11}) > Qos(WS_{21}) > \cdots > Qos(WS_{K1})$.这样,输出索引结构就建立了.在上一步确定了用户查询请求的分类后,可以进一步确定和查询输出相似度比较大的输出集合,然后根据查询输出中的 Qos需求查询出对应于各个输出的多个结果集合 $result_{o1}$, $result_{o1}$, …, $result_{ok}$ 等,将这些结果集合求交集,然后再按照用户请求中能提供的输入参数提交进行过滤就得到最终结果集合.

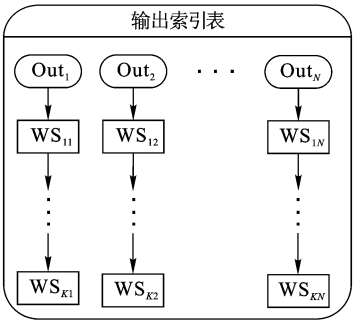

图 2 输出索引结构图

3.4 匹配算法

算法: ServiceMatching

Input: Output Services Matrix (OSM); q (Query); μ (similarity_threshold); k (Category Number)

Output: MC (Matched Services);

ServiceMatching (OSM, q, μ) {

// 建立检索系统

MC = ϕ;

SC = PLSA_CategorizingServices (OSM, k);

// 服务分类并返回类别集合 SC

For SC中的每一个类别 z Do

建立输出索引表 OI_Table;

End For

// 处理用户查询请求

计算 $P(z|q)$;

$Z_{max} = z\{z | Max(P(z|q))\}$;

// 使得 $P(z|q)$值最大的 z即为 q所属的分类,即 q匹配的分类

For请求中的每一个输出 O_i Do

For Z_{max}类别中的每一个索引 OI_j Do

If ($sim(O_i, OI_j) < \mu$) Then Do

$result_O_i$. Add (QoS_list (OI_j)); // 索引 OI_j对应的满足 QoS需求的服务列表添加到 $result_O_i$中

End If

End For

End For

$Result = \sum_{i=1}^{k} result_O_i$;

For Result中服务 S_i 的和输出匹配成功的操作 Do

If请求 q的输入包含了该操作的输入 Then Do

MC. add (S_i);

End If

End For

Return MC }

4 仿真实验

衡量 Web服务发现的基本指标为查询响应时间 (Response Time),准确率 p (Precision),召回率 r (Recall Rate)和 R 准确率 p_R.本节从这几个方面对 OntoPLSA 方法进行仿真实验.查询响应时间是指从用户发送用户查询请求到请求返回给用户花费的时间,这是对检索性能进行评估的指标. p 是指在查询结果中满足用户请求的服务数 n 占返回的所有服务数 m ($n \leq m$)的比例. r 是指查询结果中返回的满足用户请求的服务数 n 占服务库中满足用户请求的服务总数 k ($n \leq k$)的比例.由此得出: $p = n/m$, $r = n/k$, $p_R = n_k/k$, n_k 表示前 k 个返回结果中满足用户请求的服务数, k 是服务库中满足用户请求的服务总数, R 准确率更准确的反映了检索系统的准确率,它不考虑排名过于

靠后的检索结果,符合用户搜索的习惯 (用户一般只看检索结果前几页).

4.1 原始 Web服务数据集合

由于没有标准的服务发现测试集,本文从 xmethods和 WebServiceX. net等主要的 Web服务站点手工收集了 415个 Web服务组成测试集,共分为 Business, Communication, Finder等 11个类别,该测试集共含有 415个 WSDL描述文件, 2226个操作, 20259个输入和输出参数. 在 WordNet中选择 1315个待标注词汇的上位词对这 415个服务的操作名,输入输出参数进行 WordNet本体标注. 我们使用 Java语言实现了 OntoPLSA方法,实验环境为一台普通 PC机,配置 Intel E6300主频 1. 86 G, 2 G内存, Windows XP操作系统, Jdk6. 0.

4.2 性　能

为了使实验数据具有可比性,本文使用上节中的开发环境,分别实现了关键字方法 (直接关键字查询)和 Woogle算法,用来和 OntoPLSA方法做实验对比. 以收集的 415个服务为服务集做测试,分别用 3种方法做了 5次不同的查询,分别得出 5组响应时间数据,如下图 3所示.

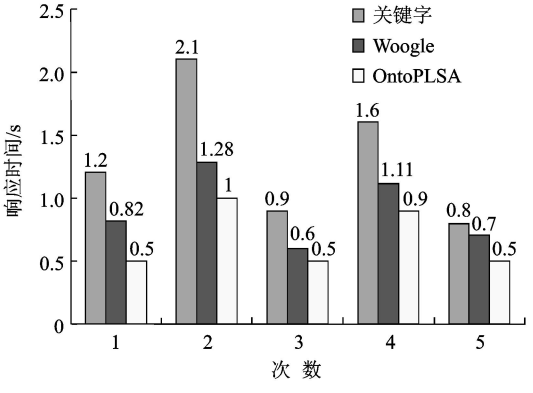

图 3　性能对比

由图 3中可以看出本文所提出的 OntoPLSA方法相比传统关键字方法和经典 Web服务发现方法 Woogle性能分别高出 48. 4%和 24. 6%,有了较大改善. 关键字方法和 Woogle方法由于没有使用语义标注来降低服务向量的维度,导致计算量相对较大,响应时间相对较长.

4.3 准确率和召回率

分别测试了平均准确率,平均召回率和 *R*准确率,并和相同条件下实现的 Woogle算法进行了对比,如下图 4所示, OntoPLSA方法比 Woogle在召回率和 *R*准确率都有所提高,但是准确率稍微有所降低 (召回率不同的条件下),这是由于对词汇使用 WordNet上位词做了语义标注,不同的词汇可能有相同的上位词,这样可以提高召回率,但是也牺牲了准确率.

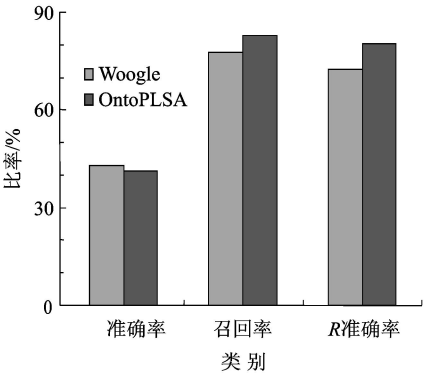

图 4　准确率、召回率和 *R*准确率

5 结 束 语

本文基于 WordNet本体标注和 PLSA提出一种语义 Web服务发现方法. 为 Web服务发现研究提供了一种方法,同时对 PLSA方法进行了改进和丰富. 实验结果表明,该方法是一种比较有效的 Web服务发现方法. 下一步工作是提高算法性能和考虑使用服务描述文档附加信息,比如参数描述等来提高检索准确率.

参考文献 (References)

[1] Bellwood T, Capell S, Clement L, et al. UDDI v3. 0 [EB/OL]. Billerica: OASIS, 2002 [2008-10-27]. http://uddi.org/pubs/uddi_v3.htm

[2] Dong X, Halevy A, Madhavan J, et al. Similarity search for web services[C]//In Proceedings of the 30th VLDB Conference (VLDB '04). Toronto: Morgan Kaufmann, 2004: 372 - 383

[3] Hofmann T. Probabilistic latent semantic indexing[C]// SIGIR '99. Berkeley: ACM, 1999: 50 - 57

[4] Miller G A. WordNet: A lexical database for English[J]. Communications of the ACM, 1995, 38(11): 39 - 41

[5] Li Y H, Bandar Z A, McLean D. An approach for measuring semantic similarity between words using multiple information sources[J]. IEEE Trans on Knowledge and Data Engineering, 2003, 15(4): 871 - 882

[6] Salton G. Automatic text processing-the transformation, analysis, and retrieval of information by computer[M]. New Jersey: Addison-Wesley, 1988

(编　辑:吴　琼)