第 34 卷 第 12 期 计 算 机 工 程 2008 年 6 月
Vol.34 No.12 Computer Engineering June 2008
·博士论文· 文章编号：1000—3428(2008)12—0025—03 文献标识码：A 中图分类号：TP393.09

基于扩展 VSM 的 Web 服务发现

陈江锋，于建军
(北京航空航天大学软件开发环境国家重点实验室，北京 100083)

摘 要：在扩展空间向量模型(VSM)的基础上设计并实现了一种 Web 服务发现机制。该机制结合 Web 服务语法和结构信息，利用相似度计算实现 Web 服务的发现。系统通过分析 Web 服务描述文档结构特点，改进现有的 VSM 模型，并加入 WordNet 扩展同义词相似概念，使得语义上等价的词被映射到相似的特征向量上，消除存在歧义的上下文，定义 Web 服务语法相似度函数，实现 Web 服务潜在语义信息的进一步挖掘。实验评测和分析表明，基于扩展 VSM 的方式使 Web 服务发现的查准率提高了 9.7%，错误率降低了 8.5%。
关键词：扩展空间向量模型；Web 服务；Web 服务发现

Web Services Discovery Based on Enhanced VSM

CHEN Jiang-feng, YU Jian-jun
(State Key Lab of Software Development Environment, Beijing University of Aeronautics and Astronautics, Beijing 100083)

【Abstract】 This paper designs and implements a Web services discovery mechanism based on enhanced Vector Space Model(VSM), combines the syntactic and structural information to calculate the similarity between services, analyzes the structural characteristics of service descriptions, adds WordNet to extend the VSM, and makes the semantically equivalent words mapped into the similar vector space which can avoid ambiguity. Similarity equation is defined to calculate the matching degree between services to discover the underlying semantics. Experiments show that the Web services discovery mechanism based on enhanced VSM improves the average precision by 9.7% and reduces the error-rate by 8.5%.
【Key words】 enhanced VSM; Web services; Web services discovery

随着Web技术的不断发展，面向服务的体系结构(Service-Oriented Architecture, SOA)被广泛接受[1]。Internet上的Web服务提供者(service provider)将应用进行封装，对外提供Web服务接口，同时在公共服务注册中心(service registry)发布Web服务描述，实现 Web服务推广。服务使用者(service requester)在公共服务注册中心上查询、发现和定位需要的Web服务，并进行绑定和调用，最终实现应用共享和组件重用。Internet上的Web服务具有广泛分布、数量庞大、功能不确定以及状态不稳定等特点。如何进行服务请求与服务能力间的匹配，最终实现Web服务发现，并满足实际应用的需要，这些是下一步研究的方向。

1 相关工作

Web服务发现是Web服务应用的前提。Web服务发现行为特点为：(1)与Web服务相关；(2)之前未知；(3)满足一定功能标准；(4)可实现资源描述机器自动处理；(5)通过匹配一系列功能或者其他标准来获取相关资源[2]。基于语法信息的Web服务发现本质上是一种利用传统信息检索机制、结合Web服务特点的新型信息检索技术。

基于语法的Web服务发现，目前主要采用基于空间向量模型 (Vector Space Model, VSM)的信息检索技术，从WSDL(Web Services Description Language)文档抽取关键词，计算不同文档之间的相似度来判断两者相似性[3]。

虽然 VSM 给出了文档与文档或关键词与文档之间的相似度，但是这种方法忽略了词与词之间的语义关系；VSM 模型认为关键词之间相互独立，但实际 Web 服务描述具有结构性特点，元素和属性值之间具有层次关系，不完全独立。现有 VSM 针对性不强，查全率和查准率都较低。

本文通过扩展 VSM 模型，提出了一种基于 Web 服务发现的方式。

2 基于扩展 VSM 模型的 Web 服务发现

在本文中，Web 服务发现主要通过计算 Web 服务请求和服务描述之间或者不同 Web 服务之间的文本相似度来获得。笔者将面向 Web 服务-Web 服务间匹配，以 WSDL 文档为例，详细介绍相关算法。

2.1 WSDL 介绍

WSDL 是一种扩展的 XML 语言，描述了分布在 Internet 环境中 Web 服务的抽象操作接口和服务实现细节。图 1 给出了 Web 服务 TravelTicketService 所表示的文档树。

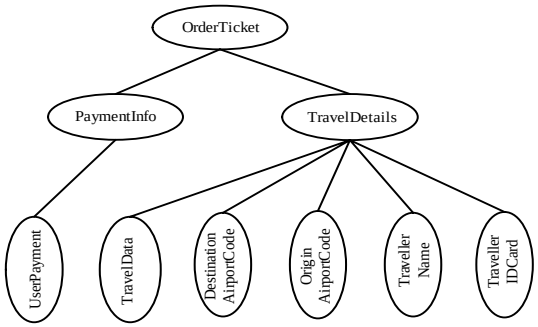

图 1 Web 服务文档树结构示例

基金项目：国家“973”计划基金资助项目“海量信息的协同性和可生存性的理论与实践研究”(2005CB321901)
作者简介：陈江锋(1976－)，男，博士研究生，主研方向：信息检索；于建军，博士研究生
收稿日期：2007-08-22 **E-mail**：chenjf@buaa.edu.cn

WSDL 使用下列元素定义 Web 服务：类型定义 types，消息 message，操作 operation，端口类型 portType，绑定 binding，端口 port 以及服务 service。这些元素及其属性共同构成了表示 Web 服务的 XML 文档树。

2.2 WSDL 文档劈分

本文将 WSDL 中的接口信息抽象为 Web 服务的功能描述。按照路径对 WSDL 进行劈分，保留原始 WSDL 文档的结构信息和元素属性值，能比较准确地表达 WSDL 的文档信息。

WSDL 文档分类或多个 WSDL 文档的比较可以转化为两个 WSDL 文档间的相似度计算，令这两篇 WSDL 文档分别为 S_a 和 S_b。每篇文档可以劈分出若干片段。

劈分片段 f 由两部分组成：路径 p 和元素属性值 v，$f=<p,v>$，WSDL 路径定义如下：

/definitions/portType/@name

/definitions/portType/operation/@name

/definitions/portType/operation/input/@message

/definitions/portType/operation/output/@message

/definitions/types/element/@name

/definitions/types/complexType/element/@name

/definitions/@name

/definitions/service/@name

/definitions/message/@name

令 F 为劈分片段集合，则 S_a 和 S_b 的劈分片段集合分别是 F_a 和 F_b。本文通过综合考虑同根词、同义词、$tf-idf$ 词频、路径的结构信息来计算劈分片断集合之间的相似度，从而表达两篇 WSDL 文档之间的文本相似度。

2.3 劈分片断相似度

劈分片断的元素属性通常是一个词组，如 TravelTicket Service 由 3 个单词构成。因此，可将其看作是一篇小型的文本文档，通过 VSM 模型来计算不同劈分片断之间的相似度，同时考虑同根词和同义词因素。

在进行相似度计算之前，本文首先对 Internet 上获取得到的 WSDL 文档进行处理，获取元素属性值词典集合，即同根词集合。

通过元素属性值拆分、有效性验证、同根词处理，删除 stop 词以及提高特殊词汇的权重等步骤，将过滤后单词存入词根集合，并统计词频、删除一些高频词、计算每一个词根的权重。

定义 1 词根权重因子 $factor_i=\lg(N/n_i+0.01)$，其中，文档总数为 N，词根集为 $LemSym$，n_i 为 $LemSym$ 中每一个词根 L_i 出现的文档数。

在计算过程中，对于劈分出的每一原始单词，从 $LemSym$ 词根集中查这个词的词根，如果词根存在，则返回词根，否则丢弃此单词。下面的词如不作特别说明，均指同根词。

同义词蕴含了 Web 服务描述的语义信息，一种最自然的方法是从领域知识的外部来源推断词的相关性。诸如 WordNet 的语义网络提供了一种获得词相似性信息的方法。WordNet 把词典中词之间的关系按照层次结构来组织分类，其中较一般的关系出现在树结构的较上层。如 spouse 出现在 husband 和 wife 的上方，即其为它们的上位词(hypernym)。可以利用 WordNet 提供的层次结构树，计算两个单词之间的语义距离。

因此，本文基于WordNet，考虑单词在WordNet词典中关联关系，包括单词之间路径的长度和深度来计算单词之间的相似度。单词之间相似度为$sim(w_i,w_j)$。

定义 2 $sim(w_i,w_j)=f(l,h)$，其中，l 指 w_i 和 w_j 之间的最短路径长度；h指 w_i 和 w_j 最大公共父节点的深度。

$$sim(w_i,w_j)=e^{-\alpha l}\frac{e^{\beta h}-e^{-\beta h}}{e^{\beta h}+e^{-\beta h}}$$

参照文献[4]实验结果，得出较优化值：$\alpha=0.2,\beta=0.6$。

劈分片断 f_a 的元素属性值 v_a 可生成 m 个词，词的集合为

$W_a=(w_{a1},w_{a2},\cdots,w_{an})$

f_b 的元素属性值 v_b 可生成 n 个词，词的集合为

$W_b=(w_{b1},w_{b2},\cdots,w_{bn})$

取词库 $W_N=W_a\cup W_b$ 共 N 个词。将 W_a 和 W_b 中每一单词映射到 N 维 VSM 空间为

$\boldsymbol{V}_a=[v_{a1},v_{a2},\cdots,v_{ai},\cdots,v_{aN}]$

$\boldsymbol{V}_b=[v_{b1},v_{b2},\cdots,v_{bi},\cdots,v_{bN}]$

对词库 W_N 中的任意两个单词 w_i 和 w_j 计算其相似度 $sim(w_i,w_j)$，当其大于某一阈值 t 时说明两个单词表达相同的概念，并根据同义词相似度来重新构建劈分片断生成的VSM向量。

考虑到Web服务描述中同义词不太可能出现在同一个劈分片断中，即如果w_i, w_j同时出现在W_a或W_b中，且$sim(w_i,w_j)>t$，仍认为这两个词表达不同的概念。因此，本文只考虑以下 2 种情况：

(1)w_j在W_a和W_b中同时出现，w_i 只在 W_a 中出现；

(2)w_i只在W_a中出现，w_j只在 W_b 中出现。

在情况(1)下，$\boldsymbol{V}_a$不变，$\boldsymbol{V}_b$变换为

$$\boldsymbol{V}_b'=\left[v_{b1},v_{b2},\cdots,\left(v_{bj}\times sim\left(w_j,w_i\right)\right)_i,\cdots,v_{bj},\cdots,v_{bN}\right]$$

在情况(2)下，则$\boldsymbol{V}_a$, $\boldsymbol{V}_b$分别变换为

$$\boldsymbol{V}_a'=\left[v_{a1},v_{a2},\cdots,v_{ai},\cdots,\left(v_{ai}\times sim\left(w_i,w_j\right)\right)_j,\cdots,v_{aN}\right]$$

$$\boldsymbol{V}_b'=\left[v_{b1},v_{b2},\cdots,\left(v_{bj}\times sim\left(w_j,w_i\right)\right)_i,\cdots,v_{bj},\cdots,v_{bN}\right]$$

对于每一对同义词都进行上述操作，并用新生成的向量 $\boldsymbol{V}_a',\boldsymbol{V}_b'$ 来计算两个劈分片断的相似度，即

$$similar\left(v_a,v_b\right)=\frac{\left\|\boldsymbol{V}_a'\boldsymbol{V}_b'\right\|}{\left\|\boldsymbol{V}_a\right\|\times\left\|\boldsymbol{V}_b\right\|}$$

2.4 Web 服务相似度

在传统 VSM 模型中，有以下假设：每一维词的出现都是独立的。WSDL 文档按照树形结构组织，其路径劈分片段也相应地包含树形结构信息。如果单纯地遵照传统 VSM 模型，就忽略了 WSDL 文档树形结构信息，因此，本文在传统 VSM 模型基础上做如下改进：考虑路径对元素属性值约束关系，结合元素属性值相似度来获取文档相似度，即 Web 服务相似度。

定义 3 $Sim_{syn}(S_a,S_b)$ 为 Web 服务相似度，即

$$Sim_{syn}(S_a,S_b)=\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}relation(p_{ai},p_{bj})\times similar(v_{ai},v_{bj})}{\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}relation^2(p_{ai},p_{bj})}\times\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}similar^2(v_{ai},v_{bj})}}$$

其中，$relation(P_{ai},P_{bj})$ 为 2 个路径表达式 $<p_{ai},v_{ai}>$ 和 $<p_{bi},v_{bi}>$ 的关系因子；$similar(v_{ai},v_{bj})$为元素属性值v_{ai}和v_{bj}的文本相似度。

relation(P_{ai},P_{bj})主要由下列因素决定：

(1)P_{ai}和P_{bj}的路径深度；

(2)P_{ai}和P_{bj}在文档集中出现的频度。

3 实验仿真与分析

本文从 Internet 环境中得到 3 500 篇 WSDL。这些 WSDL 文档中包含多个接口操作，为简单起见，本文给出的匹配算法主要针对只具有单个接口的 Web 服务。

实际 WSDL 中包含两部分数据信息：

(1)抽象接口信息，包含 Web 服务的功能特性。

(2)实现信息，与 Web 服务调用细节相关，对 Web 服务相似度的贡献不大(图 2)，因此，本文将忽略这部分信息。

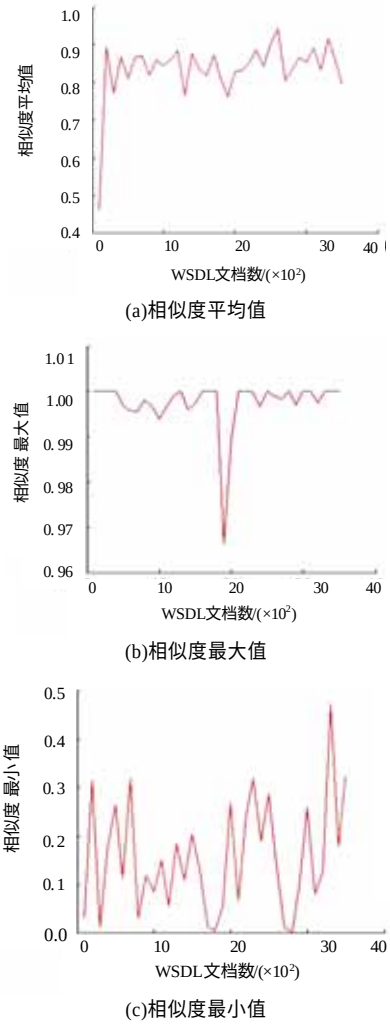

图 2 WSDL 抽象接口与原始文档相似度

3.1 性能指标

基于实际数据集，可以采用了 4 个性能指标来评测本文给出的匹配模型：错误率(error-rate)、*R*-Preciscion(*R* 个相似度排序最高的匹配结果集的查准率，其中，*R* 为较好的结果集)；Top-*N*(*N* 个最高排序中包含好的结果集的比例)；Average Precision(平均查准率)。

3.2 实验结果分析

本文给出的实验结果分为 2 个部分，在 WSDL 中只获取抽象接口部分对相似度的影响以及根据上述性能指标下的 Web 服务匹配结果的分析和评价。

从 3 500 篇 WSDL 文档抽取其抽象接口部分，并计算抽象接口部分与原始文档相似程度，结果如图 2、图 3 所示。

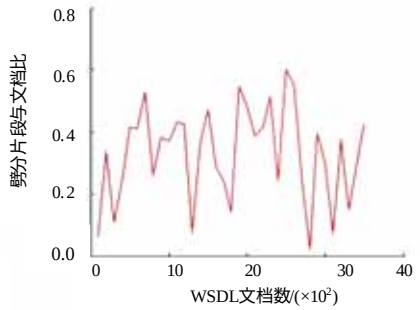

图 3 劈分片断与文档比

抽取文档抽象接口部分与原始文档之间的平均相似度为 0.836 6。抽象接口部分包含的关键词只占原始文档的 32%左右。可以认为这些抽象接口部分能够比较准确地代表整篇 WSDL 的文档信息。

通过 Web 服务匹配结果的分析，可以使用 VSM 和编辑距离(Tree Edit Distance)作为实验测试的基准。而 Web 服务相似度基于改进的 VSM 模型，指定获得最佳匹配结果的阈值，根据相似度对结果进行分类。最后根据相似度对结果进行分类。实验结果如表 1 所示，其中，EVSM 为本文提出的增强型 VSM 模型。

表 1 Web 服务匹配实验结果

方法	错误率	*R*-precision	Top5	Top10	AP
VSM	0.185 1	0.670 6	0.666 7	0.626 7	0.723 5
ED	0.408 0	0.466 5	0.626 7	0.473 3	0.647 1
EVSM	0.100 0	0.711 0	0.706 7	0.706 6	0.793 7

由表 1 实验结果可以发现，基于扩展 VSM 模型的 Web 服务发现机制在 4 个性能指标上都超过了 VSM 和编辑距离算法(ED)，取得了较高的查准率和查全率。

参考文献

[1] 李建华，陈松乔. 面向服务架构参考模型及应用研究[J]. 计算机工程, 2006, 32(20): 100-102.

[2] W3C Working Group. Web Services Architecture[EB/OL]. (2004-05-02). http://www.w3.org/TR/ws-arch/.

[3] Platzer C, Dustdar S. A Vector Space Search Engine for Web Services[C]//Proc. of the 3rd IEEE European Conference on Web Services. [S. l.]: IEEE Press, 2005.

[4] Li Y, Bandar Z A, Mclean D. An Approach for Measuring Semantic Similarity Between Words Using Multiple Information Sources[J]. IEEE Transaction on Knowledge and Data Engineering, 2003, 15(4): 871.