软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2014,25(8):1640−1658 [doi: 10.13328/j.cnki.jos.004661]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于概率主题模型的物联网服务发现*

魏 强[1,2], 金 芝[1,3], 许 焱[1,2]

[1](中国科学院 数学与系统科学研究院,北京 100190)

[2](中国科学院大学,北京 100049)

[3](高可信软件技术教育部重点实验室(北京大学),北京 100871)

通讯作者: 金芝, E-mail: zhijin@pku.edu.cn

摘 要: 针对物联网中服务数量的大规模性、服务描述的异构性以及设备服务的资源高度受限性和移动性等特点,提出了一种基于概率主题模型的物联网服务发现方法.该方法的主要特点是:1) 利用英文 Wikipedia 构建高质量的主题模型,并对类似短文本的服务文本描述进行语义扩充,使主题模型能够更有效地估计服务文本描述的隐含主题;2) 提出利用非参数主题模型学习服务文本的隐含主题,降低模型训练时间;3) 利用服务隐含主题对服务进行自动分类和文本相似度计算,快速减少服务匹配数量,加速服务文本相似度计算;4) 提出能够同时支持WSDL-based和RESTful 两种物联网服务的 signature 匹配算法.实验结果表明:与现有的物联网服务发现方法相比,该方法的准确率(precision)和归一化折损累积增益(NDCG)都有较大幅度的提高.

关键词: 物联网;服务发现;主题建模;短文本扩充

中图法分类号: TP311

中文引用格式: 魏强,金芝,许焱.基于概率主题模型的物联网服务发现.软件学报,2014,25(8):1640−1658. http://www.jos.org.cn/1000-9825/4661.htm

英文引用格式: Wei Q, Jin Z, Xu Y. Service discovery for Internet of things based on probabilistic topic model. Ruan Jian Xue Bao/ Journal of Software, 2014,25(8):1640−1658 (in Chinese). http://www.jos.org.cn/1000-9825/4661.htm

Service Discovery for Internet of Things Based on Probabilistic Topic Model

WEI Qiang[1,2], JIN Zhi[1,3], XU Yan[1,2]

[1](Academy of Mathematics and Systems Science, The Chinese Academy of Sciences, Beijing 100190, China)

[2](University of Chinese Academy of Sciences, Beijing 100049, China)

[3](Key Laboratory of High Confidence Software Technologies of Ministry of Education (Peking University), Beijing 100871, China)

Corresponding author: JIN Zhi, E-mail: zhijin@pku.edu.cn

Abstract: Internet of things (IoT) contains not only large number of services with heterogeneous description but also mobile and highly resource-constrained devices. It is key issue for IoT to find suitable services efficiently and fast. This paper proposes a service discovery approach based on probabilistic topic model for IoT. The key features of this approach include: 1) using the English Wikipedia to train a topic model with high quality and semantically enrich service text description (a form of short text) to help the topic model to extract latent topics of service more effectively; 2) employing non-parametric topic model to infer latent topics of service, which reduces the training time of the topic model; 3) making full use of the latent topics of service to automatically classify service and calculate the text similarity between service request and service, which rapidly decreases the number of services for logic signature matchmaking and accelerates similarity calculation of service text description; 4) providing a logic signature matchmaking method which supports both WSDL-based and RESTful Web service. The experimental results show that the proposed method performs much better than existing solutions in terms of precision and normalized discounted cumulative gain (NDCG) measurement value.

Key words: Internet of things; service discovery; topic modeling; short text enrichment

* 基金项目: 国家自然科学基金(61232015, 91318301)

收稿时间: 2014-01-05; 定稿时间: 2014-04-29

物联网(Internet of things,简称 IoT)是指物物相连的互联网,它将互联网扩展到物理世界中,通过射频识别(radio frequency identification,简称 RFID)、传感器、全球定位等信息传感技术,按约定的协议,把物理世界中的物体与互联网相连接,进行信息交换和通信,从而实现对物体的智能化识别、定位、跟踪、监控和管理.

为使物联网系统中的异构设备进行互相通信与协作,面向服务的架构(service oriented architecture,简称 SOA)将设备的功能封装为松散耦合的 Web 服务(称为设备服务),以一种统一的接口向外界提供,从而使各种异构的设备之间、设备与传统 Web 服务之间能够相互通信与协作.本文将设备服务和传统 Web 服务统称为物联网服务.

与传统 Web 服务相比,设备服务具有不同的特点.首先,设备服务嵌入在物理设备中,且能够提供实时数据,反映物理世界的状态,而传统 Web 服务是封装了业务功能的虚拟实体;其次,设备服务部署在资源受限的设备中(比如有限的计算和存储能力、带宽和电池等),而传统 Web 服务部署在资源丰富的计算机中;最后,由于物联网中设备的资源有限性和移动性以及无线网络自身的不可靠性,设备服务往往处于高度动态变化的环境中,服务经常消失或重新出现,而传统 Web 服务变化相对不频繁.

随着 RFID、无线传感技术、嵌入式设备和移动智能设备技术的快速发展和物理设备的廉价化,越来越多的物理设备将被连接到物联网中.如何从大量的、资源受限的异构设备中发现具有特定功能的服务来满足用户的需求,显得越来越重要.由于设备服务与传统 Web 服务相比具有不同的特点,传统 Web 服务发现方法不能有效地满足物联网服务发现的需求.首先,物联网服务数量规模较大.与传统的 Web 服务发现相比,物联网中的服务发现具有更大规模的搜索空间,需要一种高效、快速且能自动对服务进行分类管理的服务发现方法.现有的 Web 服务发现方法仅局限于小规模的服务发现.其次,设备服务具有动态的可获取性.由于无线网络自身的不可靠性、设备故障或损坏、设备的移动性以及设备的资源高度受限性等因素,服务经常消失或重新出现,即,服务具有动态的可获取性,对服务发现的实时性要求较高.现有的 Web 服务发现方法主要强调服务匹配的精确度,而未考虑匹配的实时性.第三,物联网服务具有异构的服务描述模型.由于设备种类和所具有资源量的异构性,不同的设备所提供的服务具有异构的服务描述模型,比如 DPWS (device profile for Web services)(http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01),RESTful Web 服务等.此外,物联网系统中还包括基于 WSDL(Web service description language)的传统 Web 服务,需要一种能够处理多种异构服务描述的服务发现方法.而现有的 Web 服务匹配方法仅基于 WSDL 模型或者基于 RESTful 服务模型进行匹配,缺少一种能够同时支持这两种服务的通用服务发现方法.

针对物联网服务发现的上述特点,本文提出一种高效、快速的基于概率主题模型的物联网服务发现方法.该方法利用英文 Wikipedia 构建高质量的主题模型,同时将类似短文本的服务文本描述扩充为长文本,使主题模型能够准确地估计服务文本描述的隐含主题;然后,利用非参数主题模型层次狄利克雷过程(hierarchical Dirichlet process,简称 HDP)[1]提取服务文本的隐含主题,并根据其隐含主题对服务进行自动分类,快速缩小搜索范围和加速文本相似度计算;随后,再通过计算服务与服务请求的主题相似度来进一步减小服务匹配数量;最后,提出一种能够同时支持 WSDL-based 和 RESTful 两种物联网服务的 signature 匹配方法,通过计算候选服务集与服务请求的 signature 相似度,找到与服务请求最相似的服务集合.

本文第 1 节分析物联网服务发现的相关工作.第 2 节对概率主题模型进行介绍.第 3 节详细介绍基于概率主题模型的物联网服务发现方法.第 4 节给出对比实验,并对实验结果进行分析.第 5 节对全文进行总结.

1 相关工作

目前,与物联网服务发现相关的工作主要包括探讨物联网服务发现所面临的挑战和具体的服务发现方法两个方面.Zhang 等人[2]讨论了物联网服务发现所面临的可能挑战.他们认为,物联网具有超大规模搜索空间和实时性的特点,为了支持实时搜索,可以采用基于上下文的搜索机制来缩小搜索空间,从而缩短搜索响应时间,并节省资源受限设备的能量.Valerie 等人[3]从可扩展性、服务描述的异构性以及设备服务的移动性等方面分析了物联网服务发现面临的挑战.他们认为,选取合适的服务发现架构以及对服务进行有效的分类,能够有效地解

决物联网中的大规模服务发现问题.其次,为了解决设备资源的有限性导致的服务不可获取性问题,需要设计一种高效的服务发现方法.在之前的工作中,我们分析了将传统 Web 服务发现方法应用于物联网服务发现中(特别是设备服务发现)的可行性与局限性,并认为:对于具有较大规模搜索空间、异构服务描述以及设备资源高度受限等特点的物联网服务发现而言,构建一个高效、实时的轻量级语义服务发现方法至关重要[4].

在具体的物联网服务发现方法方面,Guinard 等人[5]主要从减小设备资源消耗的角度出发,讨论了两种适用于设备服务的轻量级服务描述模型,并给出了相应的服务发现方法:首先,根据服务请求的类别确定候选相似服务集;然后,根据候选服务集的上下文信息,进一步发现与服务请求最相关的服务.但是,该方法没有考虑服务规模较大时的可扩展性和设备服务的动态可获取性问题.Teixeira 等人[6]认为:随着物联网服务数量的增加,即使执行一个简单的服务发现操作,其资源消耗都有可能超过设备所具有的资源量.他们提出采用概率发现的方法来寻找满足请求的近似服务集,以加速服务发现和减小资源消耗.Cassar 等人[7]提出一种用于物联网服务发现的混合语义服务匹配方法.该方法利用隐含狄利克雷分布(latent Dirichlet allocation,简称 LDA)[8]学习服务的隐含主题,并通过计算服务与服务请求的主题相似度确定候选服务集;然后,进一步采用逻辑 signature 匹配确定与服务请求最相似的服务集;最后,实验指出,其准确率比现有的语义服务匹配方法更好.但其存在以下几个方面的不足:1) 与互联网上的文档不同,服务描述文档类似短文本,缺乏足够的词频共现;而直接利用基于统计的 LDA 主题模型不能有效地估计出短文本的隐含主题;2) 仅采用服务语料库作为 LDA 的训练数据集,规模较小,难以获得一个高质量的主题模型,从而很难提取服务文本的真实隐含主题;3) 为了得到适用于特定应用场景的 LDA 主题模型,需要精心挑选主题参数 K,训练多个具有不同 K 值的主题模型,比较耗时;4) 没有考虑物联网中 RESTful Web 服务的匹配方法.

本文提出的方法主要结合物联网服务发现的特点,是在 Cassar 等人[7,9]工作基础上的扩充和改进.首先,为了准确地估计服务文本的隐含主题,我们提出:1) 利用 Wikipedia 对类似短文本的服务描述进行扩充,将短文本建模转换为长文本建模;2) 利用规模较大的英文 Wikipedia 数据集作为主题模型的训练集,以获得一个具有较高质量的主题模型.其次,针对 LDA 模型训练需要耗费大量时间的问题,我们提出利用非参数贝叶斯主题模型 HDP 进行服务文本主题建模,从而无需调整主题参数,大量缩短模型训练时间.最后,我们提出了能够同时支持 WSDL-based 和 RESTfull 两种物联网服务的 signature 匹配方法.

实验结果表明:与 Cassar 等人的工作相比,本文所采用的服务发现方法的准确率和 NDCG 都有较大幅度的提高.

2 概率主题模型

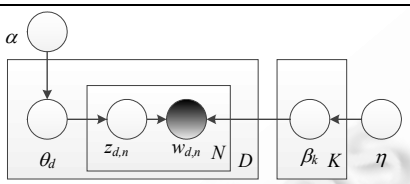
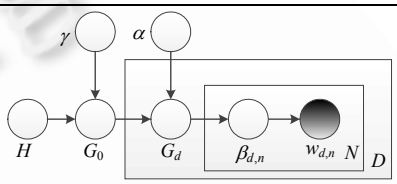
概率主题模型是一系列旨在发现大规模文档集中隐含主题结构的算法(http://www.cs.princeton.edu/~blei/topicmodeling.html).LDA 是最简单的概率主题模型,它能够提取文档的隐含主题,将文档从高维的词向量空间映射到低维的主题向量空间中.其在文本挖掘领域包括文本主题识别、文本分类以及文本相似度计算方面都有广泛的应用.

LDA 模型基于 3 点假设:1) 词袋模型(bag-of-words,简称 BOW)假设,即,LDA 认为:一篇文档是由一组词构成的一个集合,词与词之间无先后顺序关系;2) 训练文档集中文档的顺序无关紧要;3) 作为一种参数化的贝叶斯模型,训练时需预先指定主题数量 K.在 LDA 模型中,一篇文档可以包含多个主题,文档中的每个词都由其中的一个主题生成.给定特定的文档集和主题数目 K,LDA 假设文档集中所有文档共享这 K 个主题,但每篇文档具有不同的主题分布.

假设已知文档集包含 K 个主题,文档中的主题分布由参数α决定,主题中的单词分布由主题 z 和参数β共同决定.LDA 主题模型的图模型表示以及文档生成方式如表 1 左部所示,其中,带阴影的圆圈表示可观测变量;不带阴影的圆圈表示隐藏变量;带箭头的直线表示随机变量之间的依赖关系;矩形框表示重复,右下角的字母表示重复次数.对于给定的文档集,模型的训练就是估计文档集中“文档-主题”分布θ和“主题-词”分布β,可以使用 online variational Bayes(VB)方法来估计 LDA 参数[10].

Table 1 Document generation process and graphical model representation of LDA and HDP

表 1 LDA 和 HDP 模型的文档生成过程及其图模型表示

	LDA	HDP
文档生成过程	1) 对每个主题 $z\in\{1,\ldots,K\}$,抽样生成主题 z 的词分布: $\beta_k\sim Dir(\eta)$(超参数为 η的狄利克雷分布). 2) 对每篇文档 $d\in\{1,\ldots,D\}$,抽样生成文档 d 的主题分布: $\theta_d\sim Dir(\alpha)$. 3) 对文档 d 中的每个词 $n\in\{1,\ldots,N\}$,其生成过程如下: • 选择主题:从 θ_d 中抽样生成文档 d 的第 n 个词的主题: $z_{d,n}\sim Multi(\theta_d)$(多项分布); • 生成一个词:从所选主题中抽样生成词: $w_{d,n}\sim Multi(\beta_{z_{d,n}})$.	1) 对整个文档集,抽样生成主题分布: $G_0\sim DP(\gamma,H)$. 2) 对每篇文档 d,抽样生成文档 d 的主题分布: $G_d\sim DP(\alpha,G_0)$. 3) 对文档 d 中的每个词 $n\in\{1,\ldots,N\}$,其生成过程如下: • 选择主题:从 G_d 中抽样生成文档 d 的第 n 个词的主题: $\beta_{d,n}\sim G_d$; • 生成一个词:从所选主题中抽样生成词: $w_{d,n}\sim Multi(\beta_{d,n})$.
图模型表示		

虽然使用 LDA 可以成功地学习一个文档集合的主题结构,但其模型训练需要预先指定主题的数量 K.模型的质量直接依赖于主题的数量 K 的选取.该参数是一个经验值,若 K 过小,则主题粒度过粗;若 K 过大,则主题粒度过细.为了精确地估计文档中的主题个数,需要不断调整参数 K,训练多个 LDA 主题模型,比较耗时.

为了解决 LDA 模型训练参数调整问题,Teh 等人[1]提出了非参数贝叶斯主题模型 HDP.它能根据数据集自动确定主题数目 K.为了介绍 HDP,先引入狄利克雷过程(Dirichlet process,简称 DP)[1].假设 G_0 是某空间 X 上的随机概率分布,超参数 α_0 为正实数;若对空间 X 的任意一个有限的划分 $X_1,\ldots,X_r$,均有以下关系存在:

$$(G(X_1),\ldots,G(X_r))\sim Dir(\alpha_0G_0(X_1),\ldots,\alpha_0G_0(X_r)),$$

则 G 服从由基分布 G_0 和超参数 α_0 组成的狄利克雷过程,记作 $G\sim DP(\alpha_0,G_0)$.

HDP 是 DP 混合模型的多层形式,假设所有文档的主题均服从基分布 H,文档集的主题分布 G_0 服从 $DP(\gamma,H)$,文档集中文档 d 的主题分布 G_d 服从 $DP(\alpha,G_0)$,则 HDP 的形式化定义如下:

$$G_0|\gamma,H\sim DP(\gamma,H),G_d|\alpha_0,G_0\sim DP(\alpha_0,G_0).$$

其图模型表示以及文档生成方式如表 1 右部所示.与 LDA 模型训练类似,也可以采用 online variational Bayes 方法来估计 HDP 主题模型的参数[11].

与 LDA 主题模型不同的是,HDP 是一种非参数贝叶斯模型,能够从文档集中自动挑选最恰当的主题数 K.即在训练 HDP 主题模型时,K 完全由给定的数据集决定,只需训练 1 次模型即可.其次,对于未见过的新文档,HDP 能够产生新的主题,故其非常适合于对不断变化的文档集进行主题建模.而对于 LDA,即使基于现有的数据集,精心挑选主题数 K,也不能保证这些主题能够涵盖以后所有的新文档[12].

3 基于概率主题模型的服务发现

本节从 3 个方面介绍基于概率主题模型的物联网服务发现方法:1) 首先给出基于概率主题模型的物联网服务发现框架,并对该方法的思想进行简要概述;2) 然后,详细介绍如何利用概率主题模型有效地提取物联网服务描述的隐含主题;3) 最后,详细介绍服务主题匹配和逻辑 signature 匹配方法.

3.1 方法概述

基于概率主题模型的物联网服务发现框架如图 1 所示.它主要包括 3 个部分:1) 主题映射;2) 服务分类;3) 服务匹配.对每一个新服务 s,系统抽取其 signature($s_{signature}$),并通过主题映射获取其文本描述的隐含主题 s_{topics};然后,服务分类器根据 s_{topics} 自动判定其所属类别 $s_{categories}$,并将其保存到服务库中.服务库中每个服务由 4 个部分组成:1) 服务描述文档;2) 服务隐含主题;3) 服务的类别;4) 服务的 signature.

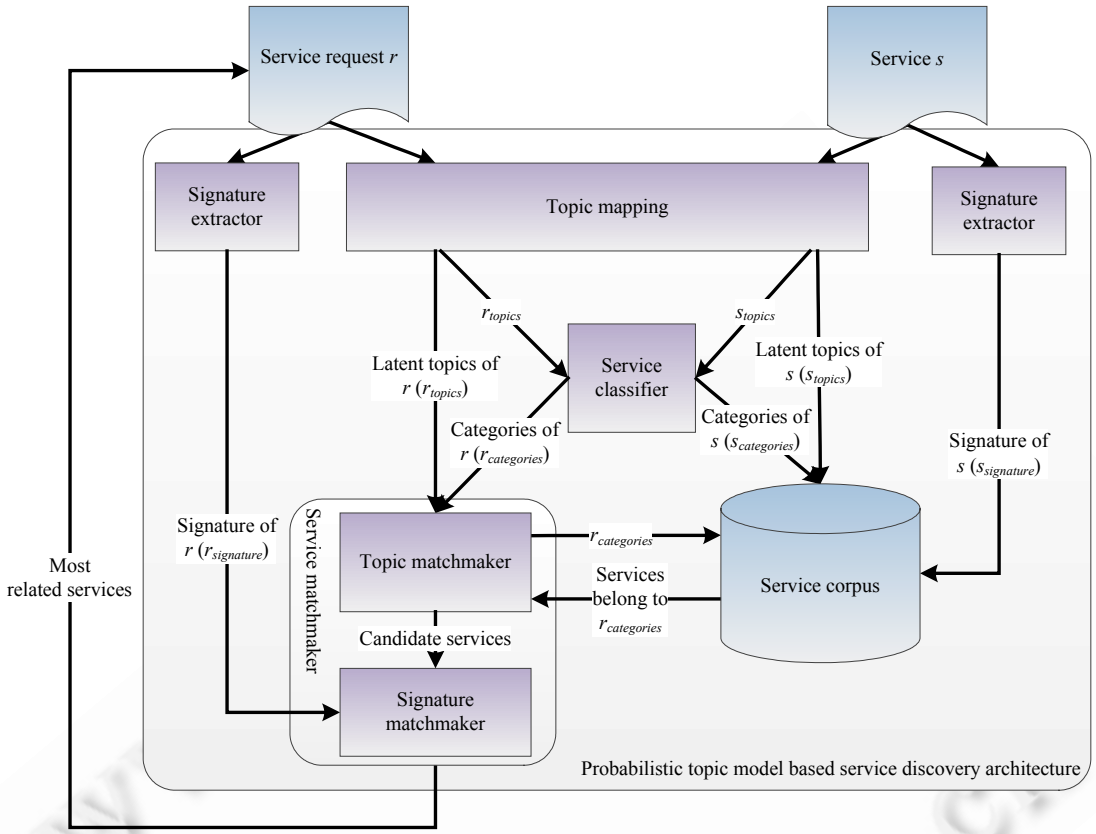


Fig.1 Service discovery architecture for Internet of things based on probabilistic topic model

图 1 基于概率主题模型的物联网服务发现框架

对于给定的服务请求 r,该方法的目的是从服务库中高效、快速地查找出与 r 最相似的服务子集.系统首先提取其 signature($r_{signature}$)和隐含主题 r_{topics},并根据 r_{topics} 自动确定其所属类别 $r_{categories}$;然后,通过主题匹配从服务库中获取属于类别 $r_{categories}$ 的服务作为初步的候选服务集,快速减小搜索范围;再从中选择与 r 的主题相似度大于一定阈值的服务作为最终的候选服务集,进一步减少服务匹配的数量;最后,将 r 与候选服务集中的每个服务进行逻辑 signature 匹配,以找到与服务请求最为相似的服务集.算法 1 对该服务发现方法进行了描述.下面重点对主题映射和服务匹配两个模块进行介绍.

算法 1. 基于概率主题模型的物联网服务发现算法.

输入:r,服务请求;θ,主题相似度阈值;α,主题相似度权重.

输出:similar Services,与 r 相似的服务子集.

1. $similarServices \leftarrow null$;
2. $r_{signature} \leftarrow SignatureExtraction(r)$;
3. $r_{topics} \leftarrow TopicMapping(r)$;
4. $r_{categories} \leftarrow ServiceClassification(r_{topics})$;
5. $services \leftarrow GetServicesFromCorpusWithSpecifiedCategories(r_{categories})$;
6. for each $s \in services$ do
7. $sim_{topic} \leftarrow TopicSimilarityCalculation(s_{topics}, r_{topics})$;
8. if $sim_{topic} \geq \theta$ then
9. $sim_{signature} \leftarrow SignatureSimilarityCalculation(s_{signature}, r_{signature})$;

10. $similarity \leftarrow \alpha \times sim_{topic} + (1-\alpha) \times sim_{signature}$;
11. Add $\langle s:similarity \rangle$ to *similarServices*
12. end
13. end
14. Sort *similarServices* by *similarity* descending;
15. return *similarServices*;

3.2 主题映射

在概率主题模型中,一个服务文本描述可视为一篇包含多个隐含主题的文档,如图 2 所示.然后,利用 LDA 或 HDP 主题模型,即可将服务文本描述从高维的词向量表示转换为低维的主题向量表示.

将概率主题模型用于服务文本主题建模的优点如下[9]:1) 互操作性:对于采用不同描述语言的服务,主题模型均可建模;2) 降维:主题模型将服务文本的高维词向量表示映射为低维的隐含主题向量,从而提高服务文本相似度计算速度;3) 自动分类:可以将服务按照其隐含主题自动进行归类.

然而,服务文本描述类似短文本,缺乏足够的词频共现,直接利用主题模型难以有效估计其隐含主题.此外,是否能够准确地估计服务文本描述的隐含主题,还取决于所训练的主题模型的质量,直接将主题模型用于物联网服务发现效果较差.

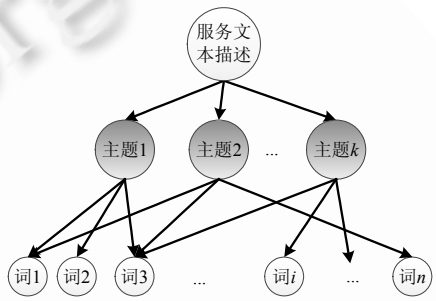


Fig.2 Topic modeling for service text description

图 2 服务文本描述主题建模

为了解决上述问题,本文一方面将类似短文本的服务文本主题建模扩充为长文本主题建模;另一方面,利用丰富的外部数据集(英文 Wikipedia)作为主题模型的训练集构建高质量的主题模型,使得主题模型能够准确估计服务文本描述的隐含主题,从而提高基于概率主题模型的物联网服务发现的准确率.如图 3 所示,利用概率主题模型精确估计服务描述的隐含主题主要包括 3 个步骤:1) 服务文本抽取;2) 服务文本扩充;3) 主题映射.下面对各个部分别进行介绍.

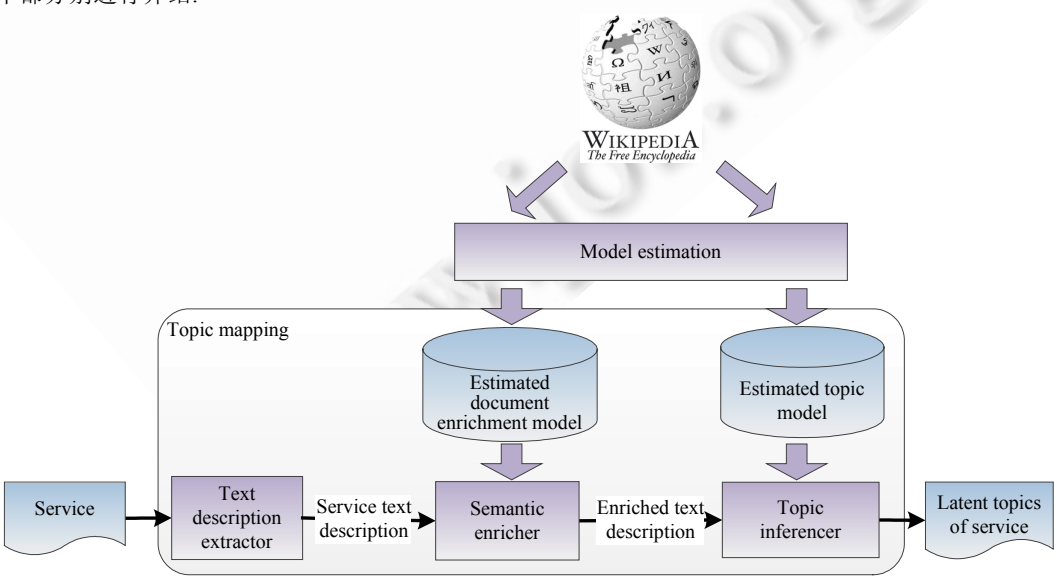


Fig.3 Topic mapping for service description

图 3 服务描述主题映射

3.2.1 服务文本抽取

根据服务所采用的描述语言,物联网服务可以分为两类:1) 基于 WSDL 的服务;2) 基于 RESTful 的服务.对于不同的服务描述形式,文本抽取需要采用不同的策略:对于采用 WSDL 描述的服务,服务名称、操作名称、操作的输入、输出参数名称、参数类型、功能性文本描述以及所引用的本体概念被抽取出来作为服务的文本;而对 RESTful 服务,我们将 HTML 标签中的文本以及所引用的本体概念抽取出来作为服务的文本.然后,对提取出的文本进行复合词拆分(比如,将复合词“wheeledcaryearrecommendedpricesoapbinding”拆分为“wheeled”,“car”,“year”,“recommended”,“price”,“soap”,“binding”),去除停用词和词干还原后,最终得到服务的文本描述可抽象表达为

$$s_{text}:\{w_1,w_2,\ldots,w_n\},$$

其中,n 表示服务文本描述中词的个数.

3.2.2 服务文本扩充

LDA 或 HDP 主题模型可以对文档主题结构进行建模,本质上均是利用词的共现信息来提取文档的隐含主题结构.然而不同于互联网上的文档,服务文本描述类似短文本,缺少足够的词频共现.直接利用 LDA 或 HDP 主题模型估计服务文本描述的真实隐含主题比较困难,虽然能够识别一些比较显著的主题,但整体效果欠佳.为了解决上述问题,本节利用 relatedness 和 word2vec 两种文本扩充方法,从英文 Wikipedia 中提取不同的特征来扩充服务文本描述,将短文本主题建模转换为长文本主题建模.下面分别对 relatedness 和 word2vec 两种文本扩充方法进行介绍.

1) Relatedness 扩充方法

Relatedness 扩充方法的基本思想是:对服务文本描述中的每个词 w_i,从 Wikipedia 中寻找与其最相关的文章集合,然后利用这些相关文章的标题(仅描述 1 个主题或概念)$C=\langle c_{i1},c_{i2},\ldots\rangle$对其进行扩展,即,将词 w_i 扩展为一个概念向量(concept vector).

Relatedness 扩充方法主要包含两个步骤:

- 构建词-概念倒排矩阵

对英文 Wikipedia 进行解析,构建 Wikipedia 词-概念倒排矩阵,见表 2.

Table 2 Wikipedia word-concept (article title) inverted matrix

表 2 Wikipedia 词-概念(文章标题)倒排矩阵

		Top L related Wikipedia concept (title of article) C			
		c_1	c_2	…	c_L
Terms in Wikipedia articles W	w_1	$tf\text{-}idf_{11}$	$tf\text{-}idf_{12}$	…	$tf\text{-}idf_{1L}$
	w_2	$tf\text{-}idf_{21}$	$tf\text{-}idf_{22}$	…	$tf\text{-}idf_{2L}$
	…	…	…	…	…
	w_M	$tf\text{-}idf_{M1}$	$tf\text{-}idf_{M2}$	…	$tf\text{-}idf_{ML}$

这样,每个词都可以用一个概念向量来表示,其中,向量中的每个值都是 TFIDF 分值,表示 term 和 concept 的相关度,其计算公式如下:

$$tfidf_{d_i}^{w_k}=\frac{(\log(f_{ik})+1.0)\times idf_k}{\sqrt{\sum_{k=1}^{t}((\log(f_{ik})+1.0)\times idf_k)^2}},$$

其中,f_{ik} 是词项 w_k 在文档 d_i 中的出现次数;idf_k 是词项 w_k 的倒置文档频率(inverse document frequency),其计算公式如下:

$$idf_k=\log(N/n_k),$$

其中,N 是文档数据集中文档的个数,n_k 是词项 k 出现过的文档个数.

在 Wikipedia 中,一个词可能会在大量文章中出现.由于相关性较低的概念对文本扩充的效果影响较小,为了提高处理效率,对每个词,我们仅选择 TFIDF 最高的前 L 个相关 Wikipedia 概念.

- 寻找相关概念

给定一个词,则可以直接利用词-概念倒排矩阵来获取其对应的 Wikipedia 概念向量对其进行扩充.比如,给定词“ontology”,“google”,与其最相关的前 10 个 Wikipedia 概念见表 3.

2) Word2vec 扩充方法

Word2vec 扩充方法的思想是对服务文本描述中的每个词 w_i,从 Wikipedia 中寻找与其最相似或最相近的词 $T=\langle t_{i1},t_{i2},\ldots\rangle$对其进行扩展,即,将词 w_i 扩展为一个词向量(word vector).

word2vec(word to vector)(https://code.google.com/p/word2vec/)是 2013 年 google 发布的基于 deep learning 的开源工具.它能将一个词转换成向量形式.通过转换,可以把对文本内容的处理简化为向量空间中的向量运算,计算出向量空间上的相似度,从而表示文本语义上的相似度.

在 word2vec 中,词向量的表示采用 Distributed Representation.其基本思想是:通过训练,将某种语言中的每一个词映射为一个固定长度的词向量(低维实数向量),将所有这些向量放在一起,形成一个词向量空间,而每一向量则为该空间中的一个点,在这个空间上引入“距离”(余弦相似度或欧氏距离),则可以根据词之间的距离来判断它们之间的相似性[13].两个词的词向量在该空间中越接近,这两个词的相似度就越高.

利用 word2vec 进行扩充主要包含两个步骤:

- 词向量的生成

将英文 Wikipedia 数据集作为 word2vec 的训练语料库,然后利用神经网络算法来训练语言模型,学习每个词的分布表示(distributed representation),即词向量,训练后即可得到语言模型和词向量矩阵(word-distributed representation matrix).本文采用 gensim(http://radimrehurek.com/gensim/models/word2vec.html)提供的开源 python 包进行 word2vec 模型的训练.

- 寻找相似词

在构建好词向量矩阵后,通过转换,即可把两个词的相似度计算简化为它们的词向量的相似度计算.比如,输入“ontology”,“google”,word2vec 找出与其最相似或最相近的前 10 个词,见表 3.

Table 3 Term enrichment example

表 3 词项扩展例子

		Relatedness (top 10 related Wikipedia concepts)	Word2vec (top 10 similar words in Wikipedia)
Term	‘Ontology’	(‘Sequence ontology’, 0.7121874) (‘Disease ontology’, 0.65095876) (‘Ontogenetic realization of categorization’, 0.59575777) (‘Standard upper ontology’, 0.59407018) (‘BRENDA tissue ontology’, 0.57258738) (‘MOD ontology’, 0.55312299) (‘Sigma knowledge engineering environment’, 0.54809589) (‘Ontology library (information science)’, 0.5466676) (‘Hozo’, 0.53641042) (‘Ontology versioning’, 0.51499021)	(‘Semantic’, 0.84863138) (‘Ontological’, 0.80479586) (‘Metatheory’, 0.78210199) (‘Relational’, 0.78010249) (‘Schemata’, 0.77560163) (‘Contextualism’, 0.77327001) (‘Logic’, 0.76600456) (‘Instantiation’, 0.7541486) (‘Enactive’, 0.74997067) (‘Wordnet’, 0.74764109)
	‘Google’	(‘Locations of google street view’, 0.64950439) (‘Google behind the screen’, 0.63834643) (‘Zingku’, 0.60227018) (‘Google the thinking factory’, 0.57014468) (‘DMarc broadcasting’, 0.52500832) (‘A google a day’, 0.5126675) (‘Google *X*’, 0.50600084) (‘Google Web history’, 0.49876775) (‘Google founders award’, 0.48546745) (‘Google business solutions’, 0.47800785)	(‘Gmail’, 0.82211536) (‘Adsense’, 0.77648675) (‘Openlayer’, 0.77452391) (‘Picasa’, 0.77356791) (‘Adword’, 0.77337897) (‘Webmail’, 0.75651157) (‘Evernote’, 0.75521922) (‘Web’, 0.75521839) (‘Skydrive’, 0.75500977) (‘Browsable’, 0.75207889)

给定服务文本描述 s_{text},对其中的每个词 w_i:

- 首先,利用 relatedness 扩充方法找到 Wikipedia 中与其最相关的前 L 个概念(相似文章的标题):

$$C_{w_i}:\{c_1,c_2,\ldots,c_L\}.$$

- 然后,利用 word2vec 扩充方法找到 Wikipedia 中与其最相似或相近的前 M 个词:

$$T_{w_i}:\{t_1,t_2,\ldots,t_M\}.$$

- 最后,利用 C_{w_i} 和 T_{w_i} 对 s_{text} 进行扩充.扩充后的服务文本描述 $s_{enrichedText}$ 为

$$\{\langle w_1,C_{w_1},T_{w_1}\rangle,\langle w_2,C_{w_2},T_{w_2}\rangle,\ldots,\langle w_n,C_{w_n},T_{w_n}\rangle\}.$$

3.2.3 主题映射

训练一个高质量的主题模型对于有效发现服务文本描述的隐含主题至关重要;而主题模型的质量,很大程度上依赖于所选用的训练数据集.为了构建一个具有较高质量的主题模型,本文采用英文 Wikipedia 作为主题模型的训练数据集.英文 Wikipedia 作为一个覆盖众多领域的大规模文本数据集,提供了足够的词频共现统计,有利于构建一个质量较好的主题模型,从而有助于准确估计服务文本的隐含主题,提高服务发现的准确率.此外,由于 LDA 主题模型的训练需要事先确定主题数目 K,而 K 是一个经验参数,需要精心调整才能得到适合特定场景的高质量主题模型,并且每次调整 K 值都需要重新训练一个新的 LDA 模型,所以需要耗费大量时间.为了解决 LDA 训练时的参数调整问题,本文采用非参数主题模型 HDP 学习服务的隐含主题.它能够根据训练数据集自动确定最合适的主题数目 K,只需训练 1 次即可,从而省去耗时的参数调整过程,大量缩短模型训练时间.

给定扩充后的服务文本描述 $s_{enrichedText}$,则可以通过训练所得的 HDP 主题模型学习出其隐含主题:

$$s_{topics}:\{topic_1:v_1,topic_2:v_2,\ldots\}.$$

同理可以获取扩充后的服务请求文本描述 $r_{enrichedText}$ 的隐含主题 r_{topics}.

算法 2 给出了具体的主题映射算法.

算法 2. 主题映射(topic mapping).

输入:s,服务.

输出:s_{topics},服务文本的隐含主题.

1. $s_{topics}\leftarrow null$;
2. $s_{enrichedText}$, $s_{text}\leftarrow TextExtraction(s)$;
3. for each $w_i\in s_{text}$ do
4. $C_{w_i}\leftarrow SemanticEnrichment(w_i,relatedness)$;
5. $T_{w_i}\leftarrow SemanticEnrichment(w_i,word2vec)$;
6. Append C_{w_i} and T_{w_i} to $s_{enrichedText}$
7. end
8. $s_{topics}\leftarrow TopicInference(s_{enrichedText})$;
9. return s_{topics};

服务的隐含主题 s_{topics} 可被用于对服务进行自动分类,为物联网服务发现建立一种有效的服务分类机制,快速缩小搜索空间并缩短服务发现响应时间,节省资源受限设备的能量.利用服务隐含主题对服务进行自动分类的具体过程见算法 3.

算法 3. 服务分类(service classification).

输入:s_{topics},服务文本的隐含主题;ω,主题分类权重阈值.

输出:$s_{categories}$,服务的类别.

1. $s_{categories}\leftarrow null$;
2. $topic\leftarrow null$;
3. $maxv_i\leftarrow 0$;
4. for each $\langle topic_i:v_i\rangle\in s_{topics}$ do
5. if $v_i>maxv_i$ then
6. $topic\leftarrow topic_i$;
7. $maxv_i\leftarrow v_i$;
8. end

9. if $v_i \geqslant \omega$ then
10. Add $topic_i$ to $s_{categories}$;
11. end
12. end
13. if $s_{categories}$ is *null* then
14. $s_{categories} \leftarrow topic$;
15. end
16. return $s_{categories}$;

3.3 服务匹配

给定服务请求 r,服务匹配的目的是从服务库中高效、快速地查找出与 r 最相似的服务子集.服务匹配主要包括主题匹配和逻辑 signature 匹配两部分.下面分别对这两部分进行介绍.

3.3.1 主题匹配

在物联网中,服务发现具有较大规模的搜索空间.此外,由于无线网络自身的不可靠性、设备的移动性以及设备的资源高度受限性等因素,设备服务具有动态的可获取性,对服务发现的实时性要求较高.为了提高物联网服务发现的实时响应性,节约资源受限设备的能量,需要快速缩小服务匹配搜索空间.

为了快速减小搜索空间,主题匹配利用服务分类和主题相似两种机制快速确定与服务请求具有相似主题的候选服务集.此外,由于逻辑 signature 匹配比较耗时,主题匹配可以快速减小逻辑 signature 匹配的服务数量,从而间接地提高了服务发现的速度.给定服务请求 r,主题匹配从服务库中快速查找出与 r 具有相似主题的候选服务集,其主要包含两个步骤:

1) 获取与 r 具有相同类别的服务子集.

在通过主题映射获取服务请求 r 的隐含主题 r_{topics} 以后,系统可以根据 r_{topics} 快速确定其所属类别 $r_{categories}$(具体见算法 3),然后仅从服务库中获取属于 $r_{categories}$ 的服务子集,快速减少需与 r 进行主题相似度计算的服务数量.

2) 计算主题相似度,确定最终候选服务集.

对于 $r_{categories}$ 中每个服务 s,可以利用余弦相似度计算其隐含主题 s_{topics} 与 r_{topics} 的相似度,记为 $sim_{topic}(r,s)$.其计算公式如下:

$$sim_{topic}(r,s)=\frac{\sum_{i=1}^{k} r_{topics}^{i} \times s_{topics}^{i}}{\sqrt{\sum_{i=1}^{k}(r_{topics}^{i})^2 \times \sum_{i=1}^{k}(s_{topics}^{i})^2}},$$

其中,k 为隐含主题个数.

若 $sim_{topic}(r,s)$大于一定的阈值θ,则将其添加到最终候选服务集中.在本文中,θ被设置为 0.5.

3.3.2 逻辑 signature 匹配

除了基于 WSDL 的传统服务以外,物联网系统中还包括大量用于资源受限设备的 RESTful 服务.本节给出一种能够同时支持这两种服务的逻辑 signature 匹配方法,逻辑 signature 匹配用于验证服务请求与服务的名称、操作(operation)名称以及操作的输入、输出参数是否匹配.

给定服务请求 r,逻辑 signature 匹配通过计算 $r_{signature}$ 和候选服务集中每个服务的 $s_{signature}$,最终确定与 r 最相似的服务子集.对于候选服务集(与服务请求 r 具有相似主题的服务)中的每个服务 s,r 与 s 的 signature 相似度计算公式如下:

$$sim_{signature}(r,s)=w_s \times sim(r.name,s.name)+w_{op} \times \frac{1}{M}\sum_{i=1}^{M}\max_{j=1}^{|s.op|}(sim(op_i^r,op_j^s)),$$

其中,$w_s+w_{op}=1$,$w_s,w_{op} \in [0,1]$;M 为服务请求 r 的操作个数;$|s.op|$表示服务 s 的操作个数;w_s 表示服务名称的相似度

权重;w_{op}表示操作的相似度权重;$sim(op_i^r,op_i^s)$为服务请求 r 与服务 s 的第 i 个操作的相似度,其计算公式如下:

$$sim(op_i^r,op_i^s)=w_{name}\times sim(op_i^r.name,op_i^s.name)+w_{in}\times\frac{1}{N}\sum_{j=1}^{N}sim(op_i^r.in_j,op_i^s.in_j)+w_{out}\times sim(op_i^r.out,op_i^s.out),$$

其中,$w_{name}+w_{in}+w_{out}=1$,$w_{name},w_{in},w_{out}\in[0,1]$;$N=\min(|op_i^r.in|,|op_i^s.in|)$,表示 r 与 s 的第 i 个操作的输入参数个数中较小的;w_{name}表示操作名称的相似度权重;w_{in}表示输入参数的相似度权重;w_{out}表示输出参数的相似度权重.

对于服务或操作的名称,它们都采用字符串进行描述,可以利用 Levenshtein 距离(http://en.wikipedia.org/wiki/Levenshtein_distance)计算服务或操作名称的相似度,其计算公式如下:

$$sim(name_1,name_2)=1-\frac{LevenshteinDistance(name_1,name_2)}{\max(name_1.length,name_1.length)}.$$

对于输入或输出参数,它们引用本体中的语义概念.为了计算它们的相似度,我们先引入本体中两个概念 c_1,c_2之间的 5 种语义匹配度:

1) Exact 匹配:c_1,c_2指向本体中的同一个概念或者两者等价,即 $c_1\equiv c_2$.
2) Plugin 匹配:c_1为 c_2的子概念,即 $c_1\sqsubset c_2$.
3) Subsume 匹配:c_1为 c_2的父概念,即 $c_1\sqsupset c_2$.
4) Intersection 匹配:c_1,c_2存在公共祖先,即$\neg(c_1\cap c_2)\subseteq\varnothing$.
5) Disjoint 匹配:c_1,c_2无公共祖先,即$(c_1\cap c_2)\subseteq\varnothing$.

为量化两个概念的相似度,我们为每种语义匹配度赋予一个权值 $sim(c_1,c_2)$:

$$sim(c_1,c_2)=\begin{cases}1, & \text{if } SemanticMatchDegree(c_1,c_2)=Exact\\ \alpha, & \text{if } SemanticMatchDegree(c_1,c_2)=Plugin\\ \beta, & \text{if } SemanticMatchDegree(c_1,c_2)=Subsume\\ \gamma, & \text{if } SemanticMatchDegree(c_1,c_2)=Intersection\\ 0, & \text{if } SemanticMatchDegree(c_1,c_2)=Disjoint\end{cases},$$

其中,$\alpha,\beta,\gamma\in[0,1]$,$\gamma$表示在本体层次结构中 c_1到 c_2的最短距离(即经过 c_1,c_2最低公共祖先的路径的长度)的倒数.在本文中,α,β被分别设置为 0.8,0.2.

此外,在计算 RESTful 服务的操作相似度时,除了考虑操作名称、输入和输出参数以外,还需要考虑操作的 HTTP 请求方法的相似度[14].RESTful 服务支持 4 种 HTTP 请求方法,包括 GET,POST,PUT 和 DELETE,它们之间的相似度度量见表 4.

Table 4 Similarity of HTTP methods

表 4 HTTP 请求方法之间的相似度

	GET	POST	PUT	DELETE
GET	1	0.8	0.2	0
POST	0.8	1	0.5	0
PUT	0.2	0.5	1	0
DELETE	0	0	0	1

最后,在计算出服务请求 r 与服务 s 的主题相似度 $sim_{topic}(r,s)$和逻辑 signature 相似度 $sim_{signature}(r,s)$以后,r 与 s 的相似度则可通过如下公式进行计算:

$$sim(r,s)=\alpha\times sim_{topic}(r,s)+(1-\alpha)\times sim_{signature}(r,s),$$

其中,α为主题的相似度权重,$\alpha\in[0,1]$.

4 实验结果及分析

本节首先介绍实验所使用的数据集以及评价标准,然后给出本文方法与其他方法的对比实验结果,并对实验结果进行相应的分析.

4.1 实验数据集

4.1.1 服务测试集

SemWebCentral(http://www.semwebcentral.org/)提供了 OWL-S,SAWSDL,hRESTS 等多种服务检索测试集,以利于评测和比较各种语义服务匹配工具的性能.

本文使用 WSMO-LITE-TC-SWRL 1.0-4g(http://seals.sti2.at/tdrs-web/testdata/persistent/WSMO-LITE-TC-SWRL/1.0-4g)(由 SemWebCentral 中的 SAWSDL 服务测试集转换而来[15])作为 WSMO-Lite 服务测试集(对于采用其他语义服务描述的服务,本文方法同样适用);由 SemWebCentral 提供的 hRESTS 服务测试集 hRESTS-TC3_release2(http://projects.semwebcentral.org/frs/?group_id=185&release_id=390)作为 RESTful 服务测试集.每个测试集都包含 1 080 个服务,39 个用于语义注释的本体文件,42 个查询及其相关服务集.其中,每个相关服务 i 具有相关度 $rel_i \in \{1,2,3\}$,其中,1 表示相关度低,3 表示相关度高.两个服务集包含相同的服务和查询,唯一的区别在于采用不同的服务描述语言.表 5 列出了测试集中不同领域的服务数量.

Table 5 Number of services for each domain in service test collection

表 5 服务测试集中各个领域的服务数量

Domain	Number of services	Domain	Number of services
Communication	58	Medical	73
Economy	358	Simulation	16
Education	285	Travel	164
Food	34	Weapon	40
Geography	60	–	–

4.1.2 Wikipedia 数据集

Wikipedia 被公认为是互联网上最丰富和全面的百科全书,涵盖了各个领域中的不同主题.在 Wikipedia 中,每篇文章仅描述一个主题或概念,且采用一个简洁的、结构良好的短语作为标题[16].

对于主题模型,Wikipedia 作为一个大文本数据集,提供了足够的词频共现统计,有利于建立一个高质量的主题模型,能够有效地提取服务文本的隐含主题.对于文本扩充方法,Wikipedia 具有丰富的词汇,有利于建立一个具有高覆盖率的词扩充模型,从而对于任意给定的服务文本,都能对其进行有效的扩充.

本文采用 2013 年 4 月的英文 Wikipedia 数据集(http://dumps.wikimedia.org/enwiki/20130403/)作为概率主题模型和短文本扩充模型的训练数据集.该数据集解压后包含 42GB,经过过滤(去除消歧页面、非文章页面以及与年表相关的页面)和预处理(去除 Wikipedia 标签、词干还原、去除停用词和低频词)以后,共包含 3 543 903 篇文章,365 307 个词.

4.1.3 评价标准

本文实验采用信息检索中两种常用的评价方式 Precision(准确率)和 NDCG(normalized discounted cumulative gain,归一化折损累积增益)作为评价标准.

Precision 是指检索返回的文档集合中相关文档的比率,其计算公式如下:

$$precision = \frac{|A \cap B|}{|B|},$$

其中,A 是相关文档集合,B 是被检索到的文档集合,$|A \cap B|$是被检索到的文档集合中相关文档的个数.

DCG(discounted cumulative gain,折损累积增益)基于两点假设:1) 高相关性的文档比边缘相关的文档要有用得多;2) 一个相关的文档的排序位置越靠后,对于用户的价值就越低.其计算公式如下:

$$DCG_n = \sum_{i=1}^{n} (2^{rel_i} - 1) \Big/ \log(1+i),$$

其中,rel_i 指检索返回的文档中排序为 i 的文档的相关性等级.

NDCG 的基本思想是:与查询越相关的文档排位越靠前,NDCG 的值就越大.其计算公式如下:

$$NDCG_n = \frac{DCG_n}{IDCG_n},$$

其中,$IDCG_n$(IdealDCG)是针对某一查询的理想的 DCG 值(最优排序).

4.2 实验结果及分析

4.2.1 实验 1. LDA 与 HDP 用于服务发现的比较

本实验对具有不同主题数目的 LDA 模型与 HDP 主题模型在用于服务发现时的准确率和 NDCG 分别进行了对比.在该实验中,LDA 和 HDP 主题模型使用 gensim(http://radimrehurek.com/gensim/)开源工具实现,并采用英文 Wikipedia 作为模型的训练数据集.其中,对于 LDA 主题模型,分别设定主题数 K=100,200,300 进行训练,得到的模型分别标记为 LDA_100,LDA_200,LDA_300.

如图 4 所示,对于两种服务,随着 LDA 主题模型主题粒度的细化,服务发现的准确率都逐渐提高;基于 HDP 主题模型的 Precision@N 和 NDCG 与 LDA 中效果最好(K=300)的模型效果大致相当;对 hRESTS 服务而言,基于 HDP 的 Precision@N 甚至比 LDA_300 的效果还要好.本质上,该实验中的 HDP 主题模型等价于主题数 K=280 的 LDA 主题模型.这在一定程度上说明服务发现的准确率不会随着 LDA 主题模型的主题粒度细化程度的增加而提高.

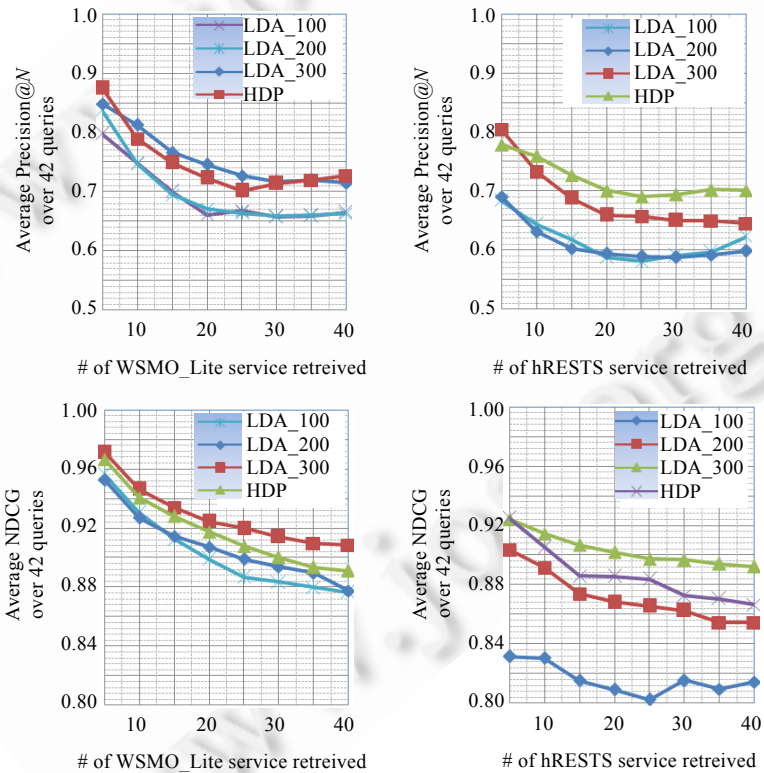


Fig.4 Comparison of LDA and HDP topic model for service discovery

图 4 LDA 与 HDP 主题模型用于服务发现的对比

对于 LDA 或 HDP 主题模型,若训练数据集较大,则需要较长的训练时间.LDA 主题模型的训练需要事先确定主题数量 K,而 K 是一个经验参数,需要精心调整才能得到适合特定场景的高质量主题模型,并且每次调整 K 值都需要重新训练一个新的 LDA 模型,所以需要耗费大量时间.而采用 HDP 主题模型,主题个数 K 可以从训练文档集中自动推出,只需训练 1 次,即可得到与针对特定场景精心训练的 LDA 模型的效果大致相当的 HDP 主

题模型,大幅度降低主题模型训练时间.

4.2.2 实验 2. 主题模型训练数据集的选取对服务发现的影响

由实验 1 可知:在用于服务发现时,无需指定主题数的 HDP 模型与需精心训练的 LDA 模型具有相似的准确率和 NDCG.故在后面的实验中,本文将使用 HDP 主题模型来进行对比实验.

Cassar 等人[7,9]采用 SemWebCentral 提供的 OWLS-TC V3.0(http://www.semwebcentral.org/projects/owls-tc)服务测试集作为 LDA 主题模型的训练数据集.它与本文所使用的服务测试集的区别仅在于所使用的服务描述语言不同.Cassar 等人指出,其实验结果比 OWLS-MX 2.0 语义服务匹配工具效果更好.本文是基于该工作的扩展与改进,所以本实验将其作为对比实验.

实验结果如图 5 所示:基于服务测试集训练的 HDP 模型用 HDP-service 表示,基于英文 Wikipedia 数据集训练的 HDP 模型用 HDP-wiki 表示,HDP+logic-servcie 和 HDP+logic-wiki 分别表示结合逻辑语义匹配的 HDP-service 和 HDP-wiki 方法.由图 5 可知:对于 WSMO-Lite 和 hRESTS 服务,HDP-wiki 的 Precision@N 和 NDCG 与 HDP-service 相比都有较大幅度的提高,这说明高质量的主题模型能够更有效地找出服务描述文本的真实主题.同时,HDP-wiki 的 Precision@N 与 HDP+logic-service 基本相当,这再次说明具有较高质量的主题模型对基于概率主题模型的服务发现的准确率影响较大.

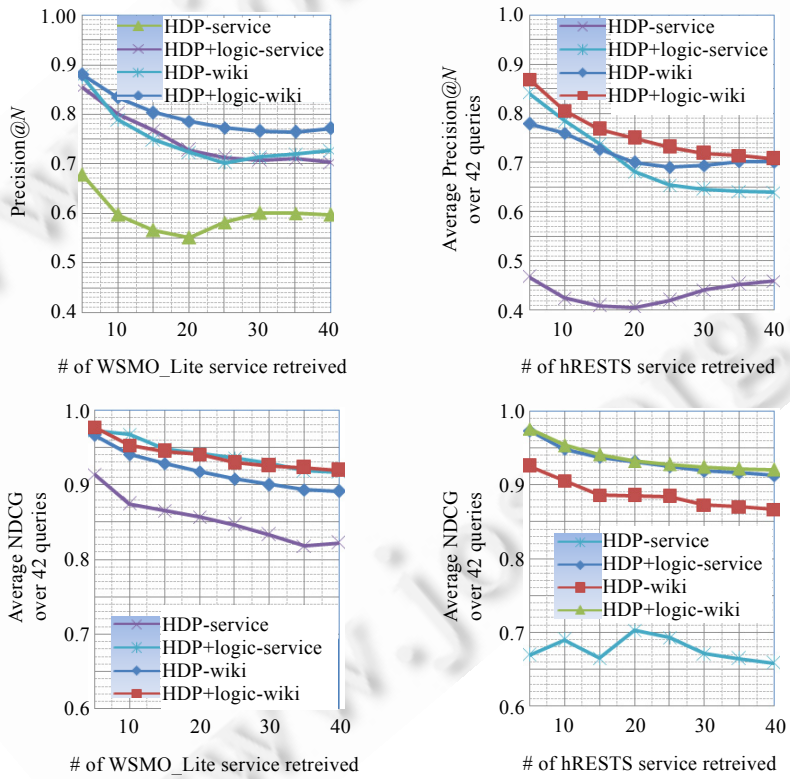


Fig.5 Comparison of different training dataset-based HDP topic models

图 5 基于不同训练数据集的 HDP 主题模型对比

此外,图 5 左上和右下两张图中均出现了拐点,但其并无特殊意义.左上图的 HDP-Service 曲线在 Precision@20(简写为 P@20)处出现拐点(P@10>P@20,P@20<P@30).其中,P@10>P@20 仅表示返回的前 10 个服务中与查询相关的服务所占的比例比第 11~20 个服务中的相关服务比例大;同理,P@20<P@30 仅表示前 20 个服务中相关服务比例较第 21~30 个服务中的相关服务比例小.右下图的 HDP-Service 曲线在 NDCG@15 处出

现拐点(NDCG@10>NDCG@15,NDCG@15<NDCG@20).其中,NDCG@10>NDCG@15 仅表示返回的前 10 个服务的相关性排序比第 11~15 个服务的相关性排序好(与查询相关性越高的服务排序越靠前,相关性排序越好,即 NDCG 值越高);同理,NDCG@15<NDCG@20 仅表示返回的前 15 个服务的相关性排序比第 16~20 个服务的相关性排序差.

4.2.3 实验 3. 服务文本描述扩充对服务发现的影响

由实验 2 可知,基于 Wikipedia 的主题模型比基于服务测试集的主题模型具有更好的服务发现准确率.故在下面的实验中,仅考虑基于 Wikipedia 数据集的 HDP 主题模型.

服务文本描述类似于短文本,缺乏词频共现;而直接利用概率主题模型估计服务文本描述的隐含主题比较困难.在本实验中,我们将探索对服务文本描述进行语义扩充是否能够有效地解决上述问题,提高服务发现的准确率和 NDCG.

本实验采用 3 种服务描述文本扩充方法,如图 6 所示,其中,

- E0 表示采用 relatedness 扩充方法,即,对于给定的一个服务文本描述 S,对其中的每个词,选择与其最相关的前 L 个概念(Wikipedia 文章标题)来扩充 S.在本实验中,L 取值为 10.
- E1 表示采用 word2vec 扩充方法,即,对给定的一个服务文本描述 S,对其中的每个词,选择 Wikipedia 数据集中与其最相似或相近的前 M 个词来表示该词,从而扩充 S.在本实验中,M 取值为 30.
- E2 表示同时采用 relatedness 和 word2vec 扩充方法.

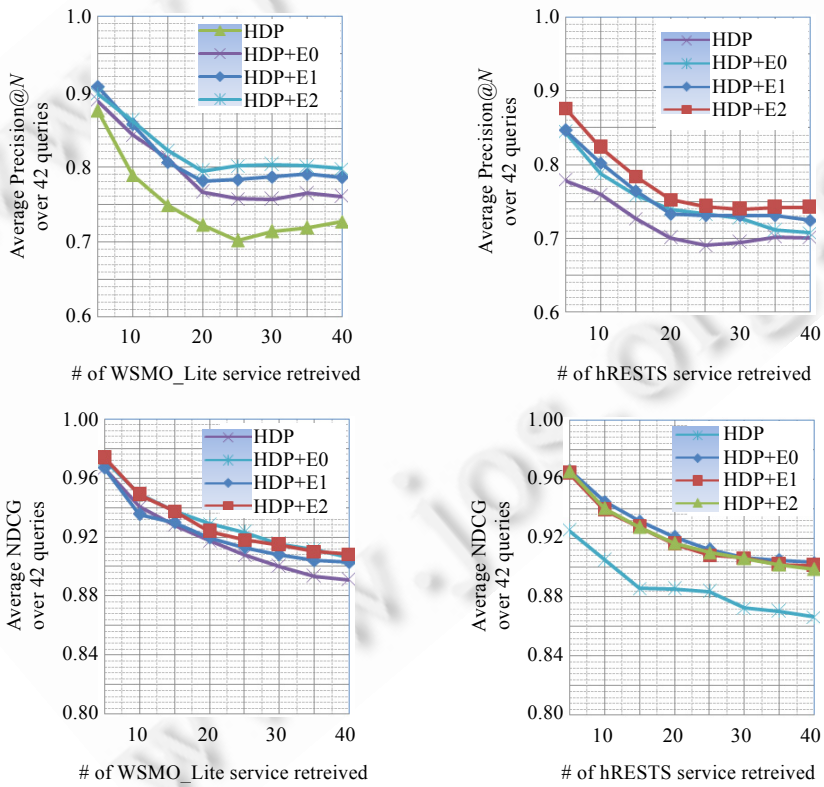


Fig.6 Comparison of different service text description enriching methods

图 6 不同服务文本描述扩充方法的比较

由图 6 可知:扩充服务文本描述后,服务发现的 Precision@N 和 NDCG 都比未扩充服务文本描述的方法效果更好.这说明对类似短文本的服务描述文本进行扩充,能够使主题模型更有效地找出服务文本描述的真实主

题,从而提高服务发现的准确率;同时,也使得与服务请求更相关的服务排位更靠前.此外,将 relatedness 和 word2vec 两种方法组合,比单独使用其中一种方法效果更好.

图 7 给出了本文所提出的服务发现方法的实验结果.对于两种服务,HDP+E2 都比 HDP+logic 的效果好,这表明对于服务发现的准确率和 NDCG 而言,扩充服务文本所达到的效果甚至比逻辑 signature 匹配更好.

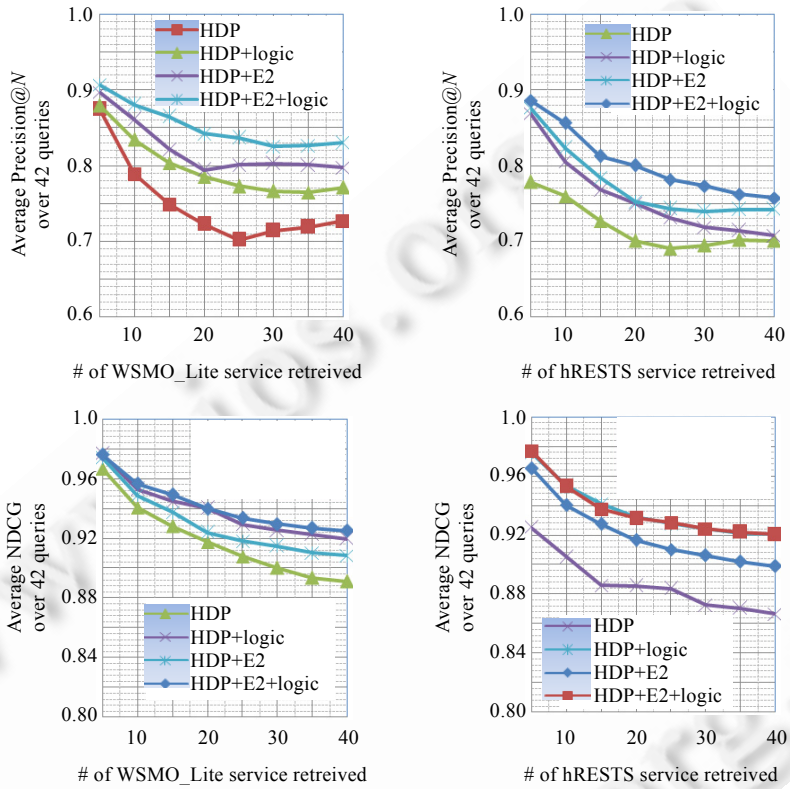


Fig.7 Service discovery based on probabilistic topic model

图 7 基于概率主题模型的服务发现

4.2.4 实验 4. 服务发现实时性分析

物联网服务具有动态的可获取性,且数量规模较大,对服务发现方法的实时性要求较高.该实验对本文提出的服务发现方法的实时性进行分析.假设服务库包含 N 个服务,则对每一个服务请求 r,服务发现方法所需的时间如下:

1) 仅采用语义 signature 匹配:$N\times T_{semantic}$.其中,$T_{semantic}$ 为计算 r 与服务的 signature 相似度的平均时间.

2) 主题模型+语义 signature 匹配:$X\times T_{topic}+Y\times T_{semantic}(Y<X,X<<N)$;其中,$T_{topic}$ 表示计算 r 与服务文本的主题相似度的平均时间,$T_{topic}<<T_{semantic}$;X 表示与 r 的文本描述具有相同主题的服务数量;Y 表示与 r 的文本描述的主题相似度大于一定阈值的服务数量.

3) 服务文本扩充+主题模型+语义 signature 匹配:$T_{annotation}+X'\times T_{topic}+Y'\times T_{semantic}(X'<<N,Y'<X)$.其中,$T_{annotation}$ 表示语义扩充 r 文本描述的时间,X'表示与 r 语义扩充后的文本描述具有相同主题的服务数量,Y'表示与 r 语义扩充后的文本描述的主题相似度大于一定阈值的服务数量.

对于包含 1 080 个 Web 服务的服务库,服务发现的平均时间(42 个查询)见表 6(机器配置为 Intel(R) Core(TM)2 Q9550 2.83GHZ,其中,程序采用单线程形式).

由表 6 可知:采用主题模型可以有效减小语义 signature 匹配的服务数量,从而大量缩短服务发现时间,能够

有效地满足物联网服务发现的实时性要求;扩充服务文本描述可以有效提高服务发现的准确率(由图 7 可知),同时,对于服务发现的实时性影响甚微.

此外,本文采用英文 Wikipedia 作为主题模型的训练集,而未采用服务集作为训练集,因此,是否触发新一轮的训练仅取决于 Wikipedia 语料库能否覆盖所有服务描述的主题,而不受服务动态变化的影响.

由于英文 Wikipedia 是目前 Web 上最大的百科全书,它覆盖了各个领域的不同主题,因此,基于 Wikipedia 训练所得主题模型能够覆盖各个服务描述的主题,只需事先训练好主题模型即可;服务的动态性主要体现在新服务出现和已有服务不可获取两个方面.当新服务出现时,只需利用主题模型识别其隐含主题,并根据其主题进行分类即可;当已有服务不可获取时,只需在其同类别的服务集中,通过主题相似和语义 signature 匹配快速查找相似服务替代即可.因此,物联网服务的动态变化对于本文服务发现方法的实时性影响较小.

Table 6 Comparison of average time of service discovery over 42 queries

表 6 42 个查询的服务发现平均时间比较

服务发现方法	WSDL (ms)	REST (ms)
语义 signature 匹配	137 670	128 250
主题模型(HDP)+语义 signature 匹配	2 304	2 081
服务文本扩充(E2:relatedness+word2vec)+主题模型(HDP)+语义 signature 匹配	4 807	2 945

5 总 结

针对物联网中服务数量的大规模性、服务描述的异构性以及设备服务的资源高度受限性和移动性等特点,本文提出了一种高效、快速的基于概率主题模型的物联网服务发现方法.该方法主要包括 3 个步骤:

1) 主题映射

由于服务文本描述类似于短文本,缺乏词频共现,直接利用主题模型估计服务文本描述的隐含主题比较困难.为使主题模型能够准确地估计服务文本描述的隐含主题,主题映射一方面利用 relatedness 和 word2vec 两种语义扩充方法对类似短文本的服务文本描述进行扩充,将短文本主题建模转换为长文本主题建模;另一方面,选取英文 Wikipedia 作为主题模型的训练集,构建具有较高质量的主题模型.

2) 服务主题匹配

利用非参数主题模型 HDP 识别服务文本的隐含主题,并根据其隐含主题对服务进行自动分类,快速缩小搜索范围和加速服务文本相似度计算;然后,通过主题匹配进一步减小服务匹配数量,确定候选服务集.

3) 逻辑 signature 匹配

它能够同时支持 WSDL-based 和 RESTfull 两种物联网服务的 signature 匹配.通过计算服务请求与候选服务集的 signature 相似度,最终找到与服务请求最相似的服务集合.

该服务发现算法的高效、快速主要体现在以下几个方面:

1) 主题模型的选取(LDA 或 HDP)

利用 LDA 训练模型需要事先指定主题数 K,而该参数通常是一个经验值,若 K 设置过大,则训练得到的主题模型主题粒度过细;若 K 设置过小,则训练得到的主题模型主题粒度过粗.对于一个特定的应用场景,需要不断地调整参数 K,训练多个具有不同主题数目的主题模型,以找到最优的主题模型.利用 HDP 则不需要调整主题参数,从而缩短模型训练时间.

2) 文本相似度的计算

通过主题映射,将服务文本的相似度计算从高维的词向量相似度计算转换为低维的主题相似度计算,从而使得文本相似度计算更快.

3) 候选服务集的选择

系统能够根据服务请求的隐含主题自动判定其所属类别,从服务库中仅提取属于该类别的服务,快速缩小服务搜索空间;然后,再选取与服务请求主题向量相似度大于一定阈值的服务作为最终的候选服务集,与服务请

求进行逻辑 signature 匹配,进一步降低逻辑匹配的服务数量.

4) 与现有物联网服务发现方法相比,准确率和 NDCG 有较大幅度的提高

基于概率主题模型的服务发现方法的准确率,主要取决于训练的主题模型是否能够准确估计服务文本描述的隐含主题.本文从两个方面来解决上述问题:

- 首先,选取 Wikipedia 作为主题模型的训练集,获得一个高质量的主题模型,极大地提高了服务发现的准确率;
- 其次,对类似短文本的服务文本描述进行语义扩充,将短文本主题建模转换为长文本主题建模,进一步提高了服务发现的准确率.

为了与已有工作进行对比,本文采用 SemWebCentral 提供的标准服务测试集.该服务测试集仅包含 1 080 个 Web 服务,规模较小.在今后的工作中,我们将把本文提出的方法用于大规模的服务测试集,以进一步测试该方法的有效性.另外,由于物联网中设备的移动性、资源高度受限性以及无线网络自身的不可靠性,设备服务往往具有动态的可获取性.今后,我们将会在本文工作的基础上,进一步考虑服务的位置、可获取时间、提供服务的设备质量等因素对物联网服务发现的影响.

References:

[1] Teh Y, Jordan M, Beal M, Blei D. Hierarchical Dirichlet process. Journal of the American Statistical Association, 2004,101(476): 1566−1581. [doi: 10.2307/27639773]

[2] Zhang DQ, Yang LT, Huang HY. Searching in Internet of things: Vision and challenges. In: Proc. of the IEEE 9th Int'l Symp. on Parallel and Distributed Processing with Application (ISPA). 2011. 201−206. [doi: 10.1109/ISPA.2011.53]

[3] Valerie I, Nikolaos G, Sara H, Apostolos Z, Panos V, Marco A, Marco AG, Amira BH. Service-Oriented middleware for the future Internet: State of the art and research directions. Journal of Internet Services and Applications, 2011,2(1):23−45. [doi: 10.1007/s13174-011-0021-3]

[4] Wei Q, Jin Z, Li G, Li LX. Preliminary study of service discovery in Internet of things: Feasibility and limitation of SOA. Journal of Frontiers of Computer Science and Technology, 2013,7(2):97−113 (in Chinese with English abstract).

[5] Guinard D, Trifa V, Karnouskos S, Spiess P, Savio D. Interacting with the SOA-based Internet of things: Discovery, query, selection, and on-demand provisioning of Web services. IEEE Trans. on Services Computing, 2010,3(3):223−235. [doi: 10.1109/TSC.2010.3]

[6] Teixeira T, Hachem S, Issarny V, Georgantas N. Service oriented middleware for the Internet of things: A perspective. In: Abramowicz W, ed. Proc. of the 4th European Conf. on ServiceWave. Berlin, Heidelberg: Springer-Verlag, 2011. 220−229. [doi: 10.1007/978-3-642-24755-2_21]

[7] Cassar G, Barnaghi P, Wang W, Moessner K. A hybrid semantic matchmaker for IoT services. In: Proc. of the IEEE Int'l Conf. on Green Computing and Communications (GreenCom). Washington: IEEE Computer Society, 2012. 210−216. [doi: 10.1109/GreenCom.2012.40]

[8] Blei DM, Ng AY, Jordan MI. Latent dirichlet allocation. Journal of Machine Learning Research, 2003,3:993−1022.

[9] Cassar G, Barnaghi P, Moessner K. Probabilistic matchmaking methods for automated service discovery. IEEE Trans. on Services Computing, 2013,PP(99):1−1. [doi: 10.1109/TSC.2013.28]

[10] Hoffman M, Blei D, Bach F. Online learning for latent Dirichlet allocation. In: Lafferty J, ed. Proc. of the Advances in Neural Information Processing Systems 23 (NIPS). 2010. 856−864.

[11] Wang C, Paisley J, Blei D. Online variational inference for the hierarchical Dirichlet process. In: Proc. of the 14th Int'l Conf. on Artificial Intelligence and Statistics (AISTATS). 2011. 752−760.

[12] Blei D, Carin L, Dunson D. Probabilistic topic models. IEEE Signal Processing Magazine, 2010,27(6):55−65. [doi: 10.1109/MSP.2010.938079]

[13] Mikolov T, Sutskever L, Chen K, Corrado G, Dean J. Distributed representations of words and phrases and their compositionality. In: Proc. of the Advances in Neural Information Processing Systems (NIPS). 2013. 3111−3119.

[14] Lampe U, Schulte S, Siebenhaar M, Schuller D, Steinmetz R. Adaptive matchmaking for RESTful services based on hRESTS and MicroWSMO. In: Proc. of the 5th Int'l Workshop on Enhanced Web Services Technologies (WEWST). New York: ACM, 2010. 10−17. [doi: 10.1145/1883133.1883136]

[15] Cabral L, Li N, Kopecky J. Building the WSMO-Lite test collection on the SEALS platform. In: Raul GC, ed. Proc. of the 9th Extended Semantic Web Conf. (ESWC). 2012. 37−48.

[16] Medelyan O, Milne D, Legg C, Witten IH. Mining meaning from Wikipedia. Int'l Journalof Human-Computer Studies, 2009,67(9): 716−754. [doi: 10.1016/j.ijhcs.2009.05.004]

附中文参考文献:

[4] 魏强,金芝,李戈,李力行.物联网服务发现初探:传统 SOA 的可行性与局限性.计算机科学与探索,2013,7(2):97−113.

魏强(1985－),男,四川仁寿人,博士生,主要研究领域为物联网服务建模与发现.
E-mail: leon.wei.cas@hotmail.com

许焱(1981－),男,博士,主要研究领域为知识工程.
E-mail: xuyan@amss.ac.cn

金芝(1962－),女,博士,教授,博士生导师,CCF 高级会员,主要研究领域为服务计算,知识工程,需求工程.
E-mail: zhijin@pku.edu.cn