基于需求功能语义的服务聚类方法

姜 波 叶灵耀 潘伟丰 汪家磊
(浙江工商大学计算机与信息工程学院 杭州 310018)

摘 要 随着互联网上服务数量的急剧增长及类型的日益多样化,如何准确、高效地发现满足用户需求的服务成为服务计算领域的一大挑战.服务聚类是提高服务发现效率的重要技术.尽管已有很多服务聚类方面的相关工作,但是现有方法不仅局限于单一类型的文档,而且鲜有考虑服务需求的功能语义.有鉴于此,文中提出一种基于需求功能语义的服务聚类方法 SCFSR(Service Clustering based on the Functional Semantics of Requirements).该方法对文档类型没有要求,且采用自然语言处理技术提取服务需求中的所有有用功能信息集;根据服务功能信息集度量服务的功能语义相似度;使用 *k*-means 算法实现服务聚类;使用 ProgrammableWeb 上 API 服务的真实数据来验证 SCFSR 方法的有效性.文中用准确率和召回率评估信息集提取的效果,并用纯度指标(Purity of Cluster)评估聚类的效果.评估结果表明,该方法可以有效地实现对服务的聚类,整个聚类的纯度达到了 57.5%,比同类方法略有提高.

关键词 服务聚类;服务需求;功能语义;聚类;服务计算
中图法分类号 TP311 **DOI 号** 10.11897/SP.J.1016.2018.01255

Service Clustering Based on the Functional Semantics of Requirements

JIANG Bo YE Ling-Yao PAN Wei-Feng WANG Jia-Lei
(*School of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou* 310018)

Abstract With the rapid growth of the number and types of services, the traditional service discovery mechanism based on keyword matching has been unable to meet people's needs due to large amount of computation. Worse still, traditional service discovery can not dig out more valuable information. So how to discover the desired services for users efficiently and accurately has become a significant challenge in service oriented computing. Service clustering, as an important technology to improve the efficiency of service discovery by reducing the scope of service search, uses the inherent feature of the service requirement text to divide services into a number of clusters. The services in the same cluster are similar to each other as much as possible while that in different types of clusters are different as large as possible. Then service clustering will map the user's request to the corresponding cluster to reduce the service search scope and service matching times. Although many service clustering approaches have been proposed, they are suitable to special types of service documents such as WSDL (Web Service Description Language) documents and OWL-S (Semantic Markup for Web Service). Worse still, it usually uses the dimensionality-reduction algorithm to filter out some features and preserve several important ones. However, at the same time, it may loose some functional semantic information. Therefore the existing approaches rarely take the functional semantics of service requirements

收稿日期:2016-12-29;在线出版日期:2017-11-02.本课题得到浙江省自然科学基金(LY15F020004)、国家自然科学基金(61202200)和浙江省科技厅公益技术研究项目(2015C33091)资助.**姜 波**,女,1970 年生,教授,硕士生导师,中国计算机学会(CCF)会员,主要研究领域为服务计算、协同计算、软件工程.E-mail: nancybjiang@mail.zjgsu.edu.cn.**叶灵耀**,男,1995 年生,硕士研究生,主要研究方向为服务计算.**潘伟丰**,男,1982 年生,副教授,硕士生导师,中国计算机学会(CCF)会员,主要研究方向为软件工程、复杂网络和智能计算.**汪家磊**,男,1991 年生,硕士研究生,主要研究方向为服务计算.

into consideration. So the focus of the current work is to cluster Web service based on functional semantics of requirements. It is difficult to extract functional semantics of requirements and compute the similarity between a pair of services. People need a more efficient way to solve these difficulties. In this paper, a novel SCFSR (Service Clustering based on Functional Semantics of Requirements approach is proposed. SCFSR is not limited to a specific type of document and can be applied to a wide range of documents. First, it uses natural language processing technologies to analyze the structure of sentences in the service requirement texts and extract the functional information in them; it filters out useless functional information to obtain the needed useful feature information which will further be lemmatized to reduce its impact on the similarity calculation. Second, based on the extracted functional information, it calculates the semantic similarity between every pair of services. Finally, according to the service similarity, it organizes the services into clusters by using the *k*-means algorithm. These obtained clusters can be further used in service discovery. API services crawled from ProgrammableWeb are used as subjects to demonstrate the feasibility of the proposed approach. We use *precision* and *recall* as two metrics to evaluate the effectiveness of the step of functional information extraction, and use *purity* as a metric to validate the effectiveness of clustering approaches. Experimental results show that the proposed SCFSR approach can cluster services effectively, and the *purity* of the final clusters is 57.5%. SCFSR is compared with one approach which does not use the functional semantics of requirements. The comparison shows that our approach is slightly better than the compared approach.

Keywords service clustering; service requirements; functional semantics; clustering; service oriented computing

1 引 言

随着面向服务的架构技术和软件即服务(Software as a Service,SaaS)的发展,服务已经成为因特网上一种重要的计算资源和软件资产[1-2].无论是基于 SOAP(Simple Object Access Protocol)的传统服务,还是流行的 RESTful 风格的服务,服务的规模、数量及种类急速增长[3-7].服务的描述文本信息仅仅在语法层面上被基于传统服务发现体系所挖掘——UDDI(Universal Description, Discovery and Integration),提取表层信息.然而在服务注册数量按指数型增长的情况下,仅支持服务描述文本的关键字匹配的服务发现方法,并不能在语义层面上挖掘更有价值的信息.人们日益增加的需求不能被当前服务发现方法所满足[7].因此,如何准确、高效地发现满足用户需求的服务成为服务计算(Service Oriented Computing,SOC)领域的难题.

服务聚类可以降低服务的搜索范围,进而提高服务发现的效率[3-6].相关研究表明,基于功能相似性进行服务的聚类可以提高服务的检索效率[5].目前,不少学者研究了基于功能相似性的服务聚类方法.文献[7]在 WSDL(Web Service Description Language)文档中运用特征选择工程,选择相关的能体现服务功能的关键特征,量化关键特征,计算服务相似度,最后聚类成簇.文献[8]统计每个词在每个服务文本出现的次数,构建词与文档的矩阵,将每个词转化成数字,然后通过生成的数字化矩阵计算服务间相似度,从而聚类成簇.但是现有工作普遍存在以下两点不足:

(1)现有的许多服务聚类方法对服务文档的类型有一定的要求:例如 OWL-S(Semantic Markup for Web Service)文档、WSDL 文档等单一类型的服务需求文档,而较少关注使用自然语言描述的没有规范标准的文档.

(2)现有的服务聚类方法较少考虑服务需求的功能语义信息:现有的方法往往通过对服务文档进行降维处理,利用空间向量模型等方法对文档进行表示和处理,进而方便度量服务间相似度,实现服务聚类成簇.

因此，如何克服传统服务聚类方法中存在的不足，提高服务聚类的性能成为服务发现中一个亟待解决的问题. 本文为克服该问题展开了研究，以服务需求语义为切入点，提出基于需求功能语义的服务聚类方法——SCFSR(Service Clustering based on the Functional Semantics of Requirements). 首先，服务需求文本需要做预处理工作，以减少噪音的影响，方便下一步的功能信息集的操作，提高功能信息集提取的准确率. 其次，在语义的层面上分析句子结构，获取相应的功能信息集. 然后，服务相似度通过功能信息集所计算得到. 即相似度越大，服务间的"距离"越近. 最后，通过 k-means 聚类算法实现服务的聚类. 以 ProgrammableWeb(PWeb)[①] 上 API 服务的真实数据为例，验证了本文方法的有效性.

本文的主要贡献如下：

(1) 提出了一种基于服务需求语义的功能信息集提取方法. 在提取过程中，我们保留了句子中所有有用的成分，而不只限于保留几个主题或者特征.

(2) 提出了一种新的服务聚类方法. 该方法对需求文本做预处理之后，从需求功能语义的角度度量服务之间的相似度，并将相似度运用到 k-means 上，聚类成簇，提高服务发现效率.

本文第 2 节将详细介绍基于需求功能语义的聚类方法，包括服务功能信息集的提取、服务相似度计算、服务的聚类算法等；第 3 节将以 PWeb 上爬取的数据为例进行实验以验证我们方法的有效性；第 4 节将介绍服务聚类的相关研究工作；最后是结论和展望.

2　SCFSR 方法

为了有效提高服务发现的效率，SCFSR 方法从经过预处理的服务需求文档中解析并筛选得到体现服务功能的功能信息集，基于提取的功能信息集计算服务间的相似度，并将相似度运用到 k-means 算法实现服务聚类. 图 1 展示了 SCFSR 方法的框架，下面各小节将详细说明框架的主要部分.

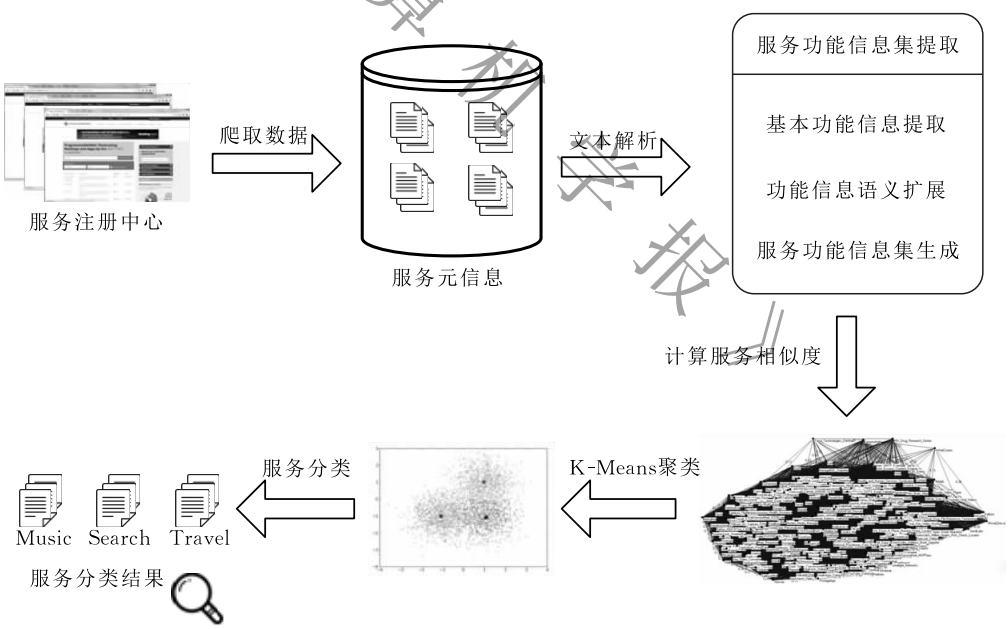

图 1　SCFSR 框架

2.1　生成服务功能信息集

服务描述文本针对于介绍服务的功能，一般被包含于服务需求文档中，帮助用户更加清楚地掌握服务特点信息. 传统的服务需求文本用自然语言进行表达[9-10].

体现用户需求的服务功能信息详细描述服务功能特点. 我们分析句子并提取服务的功能信息集. 服务功能信息的定义如下.

定义 1(服务功能信息).　$SF=(verb, object)$，用二元组的形式表示服务功能信息. 其中，$verb$ 是子句中的单个动词或者动词词组，表示服务的某个操作；$object$ 表示单个名词或名词词组，表示被 $verb$ 操作的目标.

为了让本文方法能够有效的工作，需要从服务需求中获取能够表达服务的功能信息，首先需要对

① ProgrammableWeb 网址 http://www.programmable.com

服务需求文本进行文本解析.本文通过对需求文本的分析来获取服务功能信息集,采用了开源工具 Stanford Parser① 进行处理.该工具是由 The Stanford NLP Group 开发,是目前流行的自然语言处理工具.它所具有的功能有:(1)识别单词词性,词性标注;(2)分析两单词间语法关系并生成对应相关的集合 Stanford Dependency(*SD*);(3)提取语法结构.最新版本的 Stanford Parser 包含 50 种 *SD*.

SD 可以用二元组表示,即 *sdName*(*gov*,*dep*).其中,*gov* 表示主导单词,*dep* 表示附属单词,*sdName* 则表示两个单词间的 *sd* 类型名称.例如:*SD* 为"*dobj*(*give*,*raise*)"时,语法关系 *dobj* 表示 *give* 为动词,*raise* 为名词或名词性短语作为 *dobj* 的直接宾语.

SCFSR 将服务需求文本作为 Stanford Parser 的输入得到 *SD* 集合,然后对 *SD* 集合进行分析和处理,提取出满足定义的服务功能信息集.服务功能信息集的提取算法如下所示.

算法 1. 解析并获取服务功能信息集算法.

输入:单个服务需求文本对应的 *SD* 集合
输出:文本中有用的服务功能信息集合 *NF*

```
1.  init(D); //初始化服务功能信息集
2.  FOR sd∈D //遍历集合 D 中所有 sd
3.    IF sd 符合 2.1.1 节所描述的关于两单词间的 4 种关系
4.      sf←create(sd) //创建对应服务功能信息
5.      IF sf 语义不完善
6.        esf←enrichSF(sd) //扩展基本功能信息
7.        NF←esf //保存至功能信息集中
8.      ELSE NF←sf
9.      END IF
10.   END IF
11. END FOR
12. FOR sf∈NF
13. IF sf 存在停词表 S 中的情况
14.   删除该 sf
15. ELSE 对 sf 中的动词和名词进行词性还原
16. NF←sf
17. END FOR
18. RETURN
```

上述算法的时间复杂度为 $O(N)$.其中,N 为一条服务需求语句 *SD* 集合中 *sd* 的个数.该算法为服务聚类做准备.

下面 3 小节将阐述基本服务功能信息提取、功能信息语义扩展和功能信息语义生成三个步骤的具体过程.

2.1.1 基本服务功能信息提取

基本服务功能信息只包含一个动词以及动词对应的名词.通过对 Stanford Dependencies Manual② 中定义的 50 种 *SD* 进行分析之后发现,存在几类 *SD* 可以通过简单转换直接得到基本服务功能信息,而对于那些不能直接得到的基本服务功能信息,则需要综合分析几类 *SD* 进行判断.下面是提取基本服务功能信息的四种情况:

(1)直接转换 *dobj*(*gov*,*dep*):其中,*gov* 在二元组中是动词;*dep* 是名词或名词词组,很直接地得出 *dep* 是 *gov* 的宾语.对于这类语法关系,可以直接将 *gov* 和 *dep* 作为服务功能信息中的动词部分和名词部分.例如:"This api allows you to upload your photo."包含 *dobj*(*upload*,*photo*),即可直接转换得到基本服务功能信息集{*upload*,*photo*}.

(2)直接转换 *nsubjpass*(*gov*,*dep*):在含有被动语态的句子中才能解析得到该语法关系.其中,*gov* 在二元组中是动词;*dep* 是名词或名词词组,很直接地得出 *dep* 作子句的主语.对于这类语法关系,可以直接将 *gov* 和 *dep* 作为基本服务功能信息中的动词部分和名词部分.例如:"Your photo will be uploaded to the server."包含语法关系 *nsubjpass*(*uploaded*,*photo*),即可直接转换得到基本服务功能信息集{*uploaded*,*photo*}(动词的时态会干扰相似度的计算,因此会对其进行词性还原).

(3)综合判断 *prep_p*(*gov*,*dep*)和 *nsubj*(*gov*,*dep*):当服务需求文档中动词是包含介词的词组时,我们需要的基本服务功能信息的动名词在 *prep_p* 关系中,但并不是每一个 *prep_p* 关系都能正确提取出基本服务功能信息.语法关系 *prep_p* 是一个组合 *SD*. *prep_p* 中 *p* 是一个介词,用于修饰 *gov*. *gov* 代表名词性短语或形容词,二元组中的 *dep* 是介词搭配的对象,词性是名词抑或名词性短语.由上所知,并不能简单直接分析得出对应的功能信息,需要综合 *nsubj* 语法关系推导 *prep_p* 语法关系中二元组的 *gov* 的词性是否为动词.其中,*nsubj*(*gov*,*dep*)的 *gov* 是动词.而 *dep* 是名词或名词词组,在整个句子中是主语.综合判断同一个句子中的语法关系 *nsubj* 和 *prep_p* 的 *gov* 相同时,间接获取基本服务功能信息.例如:"User can upload hotel information."包含语法关系 *nsubj*(*upload*,*user*)和

① Stanford Parser 网址 http://nlp.stanford.edu.software
② stanford dependencies manual 网址 http://nlp.stanford.edu/software/lex-parser.shtml#Download

$prep\_for(upload, information)$. 因此，可以得到基本服务功能信息{$upload, information$}.

(4) $conj(gov, dep)$判断并列的基本服务功能信息：当单个句子中存在多个服务功能信息，且服务功能信息间相互并列时，需要通过 $conj(gov, dep)$找出并列的基本服务功能信息. $conj$ 表示 gov 和 dep 存在并列关系. 例如："This api allows you to upload and share your photo and video."此句包含 $dobj$ 关系，可以直接提取出基本服务功能信息{$upload, photo$}. 然后，通过 $conj(upload, share)$和 $conj(photo, video)$找出与它存在并列关系的基本服务功能信息{$upload, video$}，{$share, photo$}和{$share, video$}.

2.1.2　功能信息语义扩展

然而，通过上述语法关系判断提取出来的信息功能集可能缺失功能语义. 例如，"User can find a cheap hotel.". 由 2.1.1 小节解析得到的基本服务信息集为{$find, hotel$}. 这就缺失了部分语义. 我们需要更加准确的信息{$find$, cheap $hotel$}. 因此，为了减少关键信息的丢失次数，有必要对部分基本功能信息集做语义扩展.

基本服务信息的语义扩展针对于对名词部分的扩展. 虽然形容词、副词、限定词等也会对名词进行修饰，但是我们分析了大量的服务需求文档中的句子中的语法关系，发现除了名词、动名词和少数形容词表达有用的语义信息以外，限定词、副词并不能体现语义信息. 因此，本文在语义扩展过程中，过滤了限定词、副词和形容词，保留名词和动名词，用来扩展基本信息服务. 这样既能很好的体现功能语义，又能减少噪音干扰. Stanford Parser 工具在进行语义扩展时考虑词与词的 nn 语法关系，即 $nn(gov, dep)$. 该二元组中的 gov 和 dep 都是名词，且二元组中 gov 被 dep 修饰.

2.1.3　生成功能信息语义

即使做完前面两个步骤生成的功能信息集，仍然有干扰词汇存在. 例如：存在一些缺乏语义的弱动词 do、let、allow 等. 因此，我们创建了一个停词表来去除包含这些动词的服务功能信息. 另外，在提取出来的服务功能信息中，由于自然语言的随意性，往往存在动词的过去时态，进行时态和名词的复数形式等等，所以我们需要对其做词形还原.

2.2　度量服务相似度

本文聚类成簇的划分标准是通过服务间相似度判断服务距离的远近. 相似度越大则服务越相似，距离越近，相似度越小则表示服务越不相似，距离越远. 计算服务相似度是聚类的准备工作. 因此，计算相似性是较为重要的一步. 本文在 2.1 节中生成了功能语信息语义相似度，其中进行了语义扩展，词形还原等步骤，获得的功能语义信息集能较好的体现服务的功能信息. 本节我们将详细介绍如何通过功能语义信息集计算服务相似度.

长短不一的服务需求文本导致不同个数的服务功能信息. 本文采用 Jacard 相似度系数(Jacard similarity coefficient)计算服务与服务间的相似度值. 如下所示：

$$J(E,F)=\frac{|E\cap F|}{|E\cup F|} \tag{1}$$

其中，E 和 F 分别是集合，集合长度不一. 计算服务相似度的具体公式如下：

$$S_s(s_1,s_2)=\frac{\sum_{i=0}^{m}S_t(t_{i1},t_{i2})}{n} \tag{2}$$

其中，定义 s_1 与 s_2 是要比较的服务，n 是服务包含的功能信息个数较多的数量，m 则为较少的服务对应的数量. $S_t(t_{i1},t_{i2})$表示计算 s_1 和 s_2 中各自服务功能信息相似度.

定义 2(服务功能信息相似度). V_1 为服务功能 t_1 包含的动词，V_2 为服务功能 t_2 包含的动词；N_{i1} 为服务功能信息 t_1 包含的名词，N_{i2} 为服务功能信息 t_2 包含的名词；w_1 对应动词所占的权重，w_2 对应名词所占的权重，且 $w_1+w_2=1$；n 是在服务功能中名词部分较多的名词数；m 是在服务功能中名词较少的数量. 计算公式如下：

$$St(t_1,t_2)=w_1\times S_{\text{word}}(V_1,V_2)+w_2\frac{\sum_{i=1}^{m}S_{\text{word}}(N_{i1},N_{i2})}{n} \tag{3}$$

其中，$S_{\text{word}}(N_{i1},N_{i2})$表示名词相似度，$S_{\text{word}}(V_1,V_2)$表示动词相似度. 单词相似度定义如下.

定义 3(单词相似度)[11]. 令 $F(c)$表示单词 c 拥有的特征集. 变量 $H(S)$代表特征集 S 信息个数，则两两单词间的相似度值计算如下：

$$S_{\text{word}}(c_1,c_2)=\frac{2\times H(F(c_1)\cap F(c_2))}{H(F(c_1))+H(F(c_2))} \tag{4}$$

其中，$H(S)$定义如下：

$$H(S)=-\sum_{t\in S}\log P(t) \tag{5}$$

其中，$P(t)$表示特征 t 出现的概率. 单词的相似度为0，表示单词间的特征集交集为空. 反之，相似度计算结果为 1 表示两个单词拥有相同特征集.

本文采用 WordNet 工具计算单词间的相似度. WordNet 是大型的英语词库，其词库中相关联了词汇和语义，便于计算相似度.

2.3 基于服务相似度的 *k*-means 聚类

服务聚类是指按照服务与服务间的相似程度，把性质类似的服务划分到同一个簇中. 相同簇的众多服务具有类似特性. 长久以来，人们为了解决不同的问题，满足不同的需求，研究并实现了各式的聚类算法. 例如，层次化聚类算法，划分式聚类算法等. 层次化聚类算法通过层次架构方式，对数据反复分裂或聚合. 而划分式聚类算法先固定初始中心，为了减少目标函数误差，反复迭代直至收敛. 其中，*k*-means 算法属于划分式聚类算法，源于信号处理的向量量化方法，现在流行于数据挖掘等领域，该算法广泛应用于系统建模的数据预处理[10,12].

在数据挖掘中，*k*-means 算法是一种聚类分析算法，不断取离中心点最近的点，减少目标函数误差至收敛. 它的中心思想是以 k 为参数，并初始化 k 个中心点，通过计算欧几里得距离来衡量点与点之间的远近. 距离越远，表示相似度越低，反之越高. 本文计算服务功能信息集衡量服务间的相似度程度，并运用到 *k*-means 算法中. 算法如下所示.

算法 2. 基于服务相似度的 *k*-means 算法.

输入：NF、k //NF 预处理后的服务功能信息集，k 为簇的个数

输出：k 个簇

1. Init(C_i)(i=0,…,k) //初始化准备
2. *centers*=SelectCenter(NF,k) //*centers* 是存储中心的数组. 从 NF 功能信息集选择 k 个作为中心点；
3. WHILE(*result*!=*formerResult*){//*result* 为当前聚类结果，*formerResult* 为上一次聚类结果
4. *similarity*[k]=Sim($s\in NF$,*centers*[k]) //计算单个服务到中心点的距离，并存储到数组中
5. max(*similarity*[k]) //选择最大相似度，并划分到相似度最大的类簇
6. ReSet(*centers*) //重新选择中心点
7. }
8. RETURN *result*; //返回 k 个类簇

上述算法的时间复杂度为 $O(I*N^2*k)$. 其中，I 是迭代次数，k 要聚类的个数，I 和 k 可以当成是常数，N 是服务相似度的个数. 因此，该算法的时间复杂度为线性的 $O(N^2)$. 结合算法 1 计算服务的相似度，将算法 1 的输出用作算法 2 的输入，输出服务的类簇.

3 实例分析

为了展示本文方法的具体实现过程，并验证其有效性，本节将使用 PWeb 网站上 API 服务的真实数据作为实例进行实验和分析. 本实验的实验环境包括 Win7 64 位操作系统，JDK8.0，内存 4GB，主频 3.1GHz.

3.1 数据集

PWeb 为容纳了大量不同领域 API 信息的服务注册平台[6]. 截止目前，网站上共罗列了 10000 多个 API 服务. 如图 2 所示，PWeb 上每个 API 信息提供了详细的内容，包括 API 提供者、主页、分类信息、标签等. 另外，若该 API 已经废弃，则在该网页内有相应显示. API 服务的描述信息往往都是 100 多个英文单词的文本，包含了关于服务功能信息的描述，可以看成是 API 服务的需求文档. 因此，PWeb 上的 API 服务满足本文工作的要求. 下面将以 PWeb 上的数据作为实例进行研究.

图 2 PWeb 上 API 服务的元信息

为了实现数据获取，我们用 Python 编写了一个爬取网站 API 数据的脚本. 首先通过脚本模拟浏览器发送 URL 请求，其次定位到具体的 API 信息界面，通过正则表达式匹配网页中关键信息，并对爬取的信息做预处理. 最后将处理过的信息存入数据库，等待进一步分析[9]. 在数据收集过程中，出现了

以下问题：(1) 服务描述为空或者分类标签为空；(2) 相同的服务反复提交注册. 针对问题(1)，关键信息为空的服务给予剔除. 对于问题(2)，在爬取过程中放入 Set 集合，可保证服务不重复. 最终，我们的数据集包含 15 928 个 API 服务.

此外，PWeb 上的服务信息是由用户自己提供，存在一定的随意性和不规范性：(1) 存在一些特殊字符，例如："/"、"€"等；(2) 存在语句语法问题. 上述问题都可能影响功能信息集的提取准确率. 因此，在对描述文本进行服务功能信息提取之前，我们进行了一定的预处理工作：(1) 对特殊字符进行去除或者替代，如：将"&"替换成"and"，"/"替换成"or"，去除类似"€"这样不影响语义的特殊字符；(2) 改进语法规范，例如：句子首字母进行大写等.

3.2　实验过程及结果

3.2.1　服务功能信息集提取

通过第 2 节的功能信息集的提取和处理之后，最终在 15 928 个 API 服务中获得了 15 653 个不为空的功能信息集. 如图 3 所示，展示了功能信息集的提取状况. 由于自然语言描述的服务需求存在随意性和不规范性，文本太短或者为空都不能提取对应有用的信息功能集. 其次其需求文本的语法格式不正确，缺少句子的某些成分，主谓宾不完整，也不能被正确解析. 对于解析出服务信息集为空的情况，我们给予剔除.

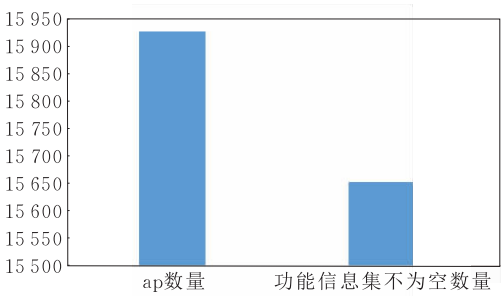

图 3　API 数与功能信息集数

3.2.2　服务聚类

在 3.2.1 节里提取出所有的功能信息集不为空的服务中，我们统计数据库中各个类别的 API 的数量，选取了 3 个数量平均的 API 类别，分别是 Medical(132 个)，Email(263 个)，Video(255 个)，总共 API 数量为 650 个. PWeb 上已有的分类的个数差别很大，虽然有最高数量的分类 1041 个，但是属于个别现象. 基本上各个分类分布在 10 个左右数量的 API 服务. 本文选择数量规模相对均匀的分类集合的原因是各种类服务 API 数量相差太大，会影响实验分类结果，影响分类准确率. 因此，我们选取分类数据标准是在数量相差不大的情况下，随机选取 3 个类别当做我们的实验数据. 同时，令聚类结果与原有标签上的分类做比较，并验证实验的有效性.

对于运用 Stanford Parser 开源工具获得的功能信息集，做了词形还原等操作减少噪音，并用 WordNet 计算两个服务之间的相似度. 如图 4 所示，一个点表示一个服务，描述了各个服务之间相似度程度. 其中，两个服务相似度最高的 API 名字分别叫作 Sociagram，Socialca. 经分析其需求文档和提取的信息功能集发现 Socialca 服务是 Sociagram 服务的改进版本，其需求描述文本都是介绍视频播放的功能. 服务提取的功能信息集越相似，则相似度的数值越高. 服务之间没有关系，则相似度为 0.

本文设定聚类个数 $k=3$，即输出结果有 3 个类别，如图 5 所示.

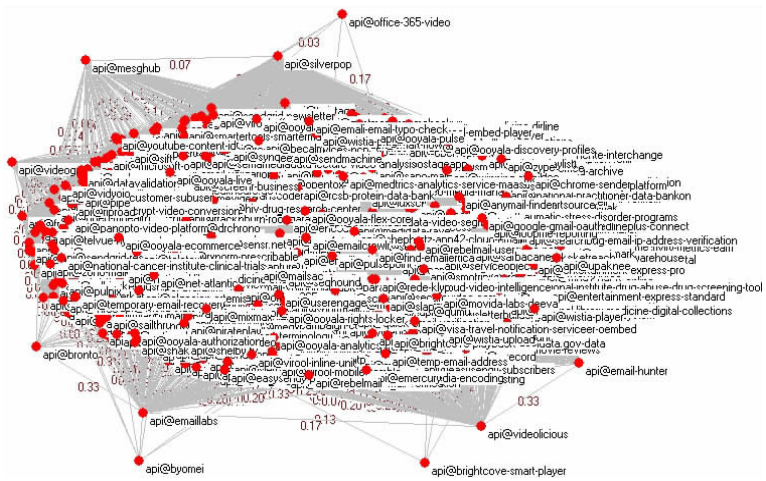

图 4　服务间相似度值

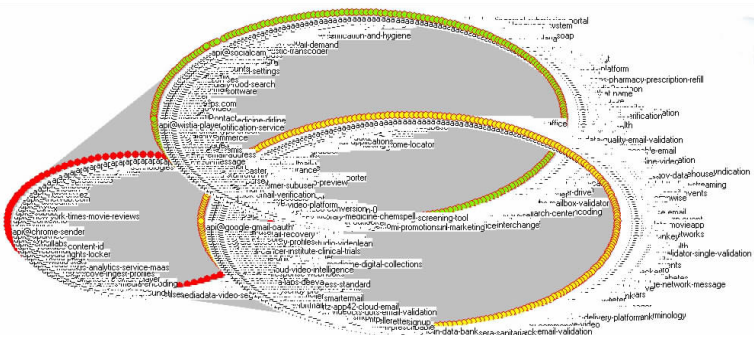

图 5 服务聚类结果

表 1 可以清楚地看出 3 个类别的数量分别是 276,265 和 109. 三个分类在数量上跟原有分类的数量稍微有所不同. 其三个分类的中心点所在的类别都不相同,分别是 Email,Video 和 Medical. 同时表 1 列出了各个分类中服务类别数量组成情况. 为了验证是否能有效地提取服务信息功能集,是否能够正确地进行聚类,我们对此进行了分析. 表 2 列举了每个分类的中心点和中心点对应的服务功能信息集. k-means 聚类收敛后的中心点在一定程度上可以当做是一个簇的类别情况. 从表 2 中分析可以得到,中心点为 Sendy 的服务,其功能信息有{send email,manage email…}等等,都是跟 Email 状况有关. 中心点为 Viddler 的服务,其功能信息集有{upload video,create video…},基本上都是针对视频操作. 中心点为 Ihealth 的服务,其基本信息集{track heart data…},基本上都是跟医疗健康相关.

表 1 聚类结果及服务构成

聚类结果	中心点服务	服务数	分类构成		
			Email	Video	Medical
分类 1	Sendy	276	166	51	59
分类 2	Viddler	265	98	125	42
分类 3	Ihealth	109	21	16	71

表 2 聚类中心点与服务功能信息集

中心点服务	原有分类	功能信息集
Sendy	Email	{send newsletter email, send list, manage newsletter email, export subscribe email, import email, allow developer, create email}
Viddler	Video	{update data, upload video, generate token, create video, create video, delete comment, get video detail, display video, delete video, edit video, start video, move video, rename video}
Ihealth	Medical	{provide pressure monitor, track health data, track heart rate, trick blood pressure, interact user, use module information}

对服务功能描述不全或者描述含糊不清的服务需求文档,我们分析了多个类似的需求文档发现:根据语义层面的相似性,这些服务会被划分到相似度最高的类别之中. 例如:Campaign Monitor 服务,在 PWeb 提供的标准分类为 Email,但需求文档提取的服务功能信息并没有体现有关 Email 邮件收发的操作,更多的体现测量服务. 因其语义层面上更加靠近分类 3,将其归类到分类 3. 服务 Aetna CarePass 的功能信息集体现了大量的数据访问,数据交换等相关的功能,因此由原来的分类 Medical 分到分类 1 中.

本文采用纯度指标(Purity of Cluster)[13] 来评价聚类的有效性. 纯度定义如下.

定义 3(聚类纯度)[13]. 令 D 为服务集合,C 表示 D 上多次聚类结果中的一次聚类结果,$C_k \in C$ 指的是一个聚类结果中第 k 个簇,S 为 D 上的原有的标准的聚类结果,$s \in S$ 指的的一个簇的标准聚类结果,则 C_k 类的聚类纯度 $PC(C_k)$ 定义为

$$CP(C_k)=\frac{1}{|C_k|}\max(|C_k^s|) \tag{6}$$

其中,$|C_k|$ 表示 C_k 类包含服务的数量,$|C_k^s|$ 代表标准簇 s 与聚类 C_k 簇中服务的交集的个数. 本文按照 PWeb 提供的服务分类结果作为标准簇. 因此,整个服务聚类结果的纯度 $CP(C)$ 定义为

$$CP(C)=\sum_{k\in C}\frac{|C_k|}{|D|}CP(C_k) \tag{7}$$

聚类结果的准确性由聚类纯度表示,越高的聚类纯度反映越好的聚类效果. 表 3 给出了各类的纯度及整个聚类的纯度.

表 3 API 服务聚类结果纯度

聚类结果	纯度/%
分类 1	60.0
分类 2	47.2
分类 3	65.3

本文使用斯坦福大学的解析工具分析服务需求文档的语法关系，提取相关的功能信息集. 由于自然语言的随意性，导致提取的功能信息集的准确度对服务聚类有一定的影响. 再者，由于自然描述的需求文档存在功能与描述不符合的情况. 上述情况都可以影响聚类效果. 服务聚类的结果表明：本论文通过提取服务功能信息集，并通过计算服务功能信息集的相似度，并将相似度当做衡量服务的“距离”，可以很好地从需求语义角度进行聚类.

3.3　实验结果分析

3.3.1　实验效果评估

过滤掉功能信息集为空的服务集合之后，功能信息集的提取情况如图 6. 由于服务需求文档比较简短，描述语句个数基本在 3～5 句之内，平均单词数量也在 60 个左右，再加上过滤了没有体现服务功能信息的功能信息，大部分的服务对应的功能信息集在 10 个以内.

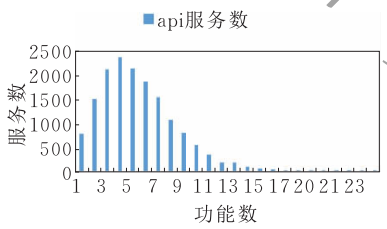

图 6　服务功能信息提取情况

本文在 15928 个服务中随机选取了 600 个服务作为实验数据验证服务功能信息提取算法的有效性. 我们构建一个网络平台，让 30 名研究生通过在网络平台上获取实验数据，用手工方式对描述文本进行功能信息提取，然后再对实验数据进行自动提取. 从而得到不同的功能信息集. 我们将汇总 600 条对应的服务功能信息. 由于每个学生对句子的理解程度不同，因此有可能有不同的标准，但是大体上差不了太多，我们忽略这个小问题. 通过计算准确率和召回率评估提取效果：

$$precision = \frac{|N_{FM} \cap N_{FU_i}|}{|N_{FM}|} \tag{8}$$

$$recall = \frac{|N_{FM} \cap N_{FU_i}|}{|N_{FU_i}|} \tag{9}$$

其中，N_{FM} 定义为算法分析句子提取的信息集，N_{FU_i} 定义为第 i 个研究生人工按照自己标准解析得到的服务功能信息集. 表 4 汇总了服务功能信息集提取的情况.

表 4　服务功能信息提取结果评估

服务功能信息集	平均准确率	平均召回率
N_{FU_1}	0.715	0.984
N_{FU_2}	0.694	0.952
N_{FU_3}	0.768	0.965

通过表 4 展示的数据，发现不同的服务提取的服务信息集的准确率不同，原因在于每个人都有每个人的标准，并且由于自然语言的随意性和自由性，导致每个人理解不同. 召回率基本上在 98% 左右，接近于 1. 这表明本论文的提取功能信息集的算法能够很好的工作，提取的信息集体现了大部分的语义功能. 由于在提取过程中还是没有完全过滤体现功能语义的信息集，所以仍然包含无意义的功能信息，影响提取的准确率.

3.3.2　参数对聚类结果的影响

本论文的方法含有三个参数：k，w_1 和 w_2. 其中，k 是要聚类的个数，w_1 是语义提取之后计算相似度的动词权重，w_2 是计算相似度的名词权重，且 $w_1 + w_2 = 1$.

图 7 展示了 k 和 w_1 对纯度影响的三维图，图 8 和图 9 分别显示了 k 和 w_1 对纯度影响的二维图.

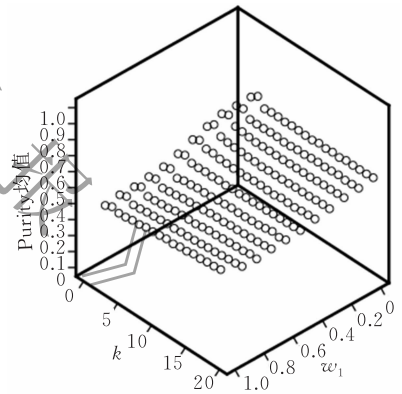

图 7　k 和 w_1 对纯度的影响

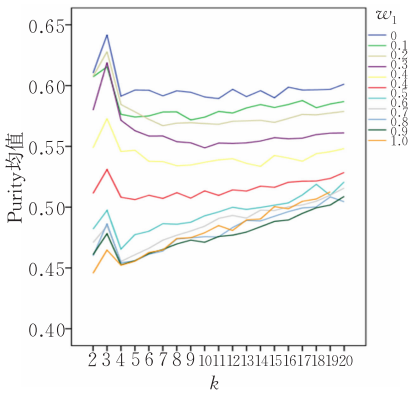

图 8　k 对纯度的影响

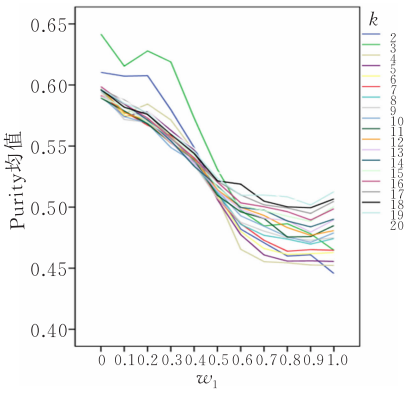

图 9 w_1 对纯度的影响

图 8 中 k 的取值的范围是 2 到 20. 从图 8 可知，在 $k=3$ 的时候，方法的效果较好，跟 3.2.2 节所述相呼应. 从图 9 可知：在 k 值固定的情况下，随着 w_1 的变小，聚类纯度越高. 从图 9 可知，当 $w_1=0$ 的时候，纯度最高. 这说明提取服务功能信息集中，只提取名词部分有益于提高聚类效果.

3.3.3 相关比较

本小节将本文提出的 SCFSR 方法与没有基于功能语义的算法进行比较，验证 SCFSR 方法的有效性.

尽管已有很多服务聚类方面的工作，但是我们无法跟现有工作进行充分的比较，主要原因如下：

(1) 我们无法获取现有工作中所提到的数据集，现有工作中也未详细描述他们获取数据的所有细节，我们无法获得规模一致的数据集.

(2) 现有工作也并未提供其方法程序源码的获取方式. 同时，现有工作所提出的方法中很多细节描述并不清晰，我们无法原样复现他们工作中的方法.

综上所述，我们无法将本文提出的 SCFSR 方法与别人的方法进行充分比较. 在本文中，我们实现了基于文本的相似度提取算法(该方法没有 2.1 节所述步骤)，直接对文本做余弦相似度计算，并且用 k-means 进行聚类($k=3$). 图 10 显示了两个方法的对比结果. 从图中可知，本文提出的基于功能语义的算法效果更好.

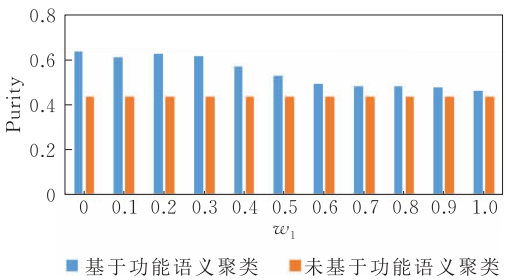

图 10 SCFSR 与未基于功能语义的方法对比

4 相关研究

服务聚类将机器学习算法运用到服务发现中，目的是提高发现效率的有效技术. 通常使用服务聚类算法对数据进行预处理，为其他复杂服务匹配算法做准备. 挖掘句子更深层次的语义信息，把类似的服务聚成类簇. 再者，通过服务发现机制，将用户的搜索请求定位到相应的服务类簇，从而降低服务的搜索数量和匹配计算量，提高服务发现效率. 目前，对于服务聚类国内外已有大量研究.

文献[7]从规范的 WSDL 文档获取固定几个能够体现服务功能信息的词语，通过这几个关键词语将相似的服务划分到一块. 文献[8]统计每个词在每个服务文本出现的次数，构建词-文档的矩阵，将每个词转化成数字，然后通过生成的数字化矩阵服务间计算相似度，从而聚类成簇. 这类聚类方法采用对文档进行降维处理，构造稀疏矩阵. 但预处理得到的文档还是没有提出相关噪音. 并且由于稀疏矩阵有很多维数导致计算量庞大且复杂. 文献[14]基于 Agglomerative 算法，提出 SWSC 的 Web 服务聚类算法，该算法把具有类似特点的服务划分在一块. 该算法都基于 Agglomerative 层次式算法的思想，复杂度较高，耗时长. 文献[15]提出了统计聚类方法，用于 Web 服务的分布式向量空间搜索引擎，可以处理非常大的服务存储库，提高服务发现效率.

文献[16]基于规范 WSDL 文档相似的情况下，将 Tags 当做服务聚类的关键特征，再结合聚类算法对服务进行聚类. 文献[17]深入研究了结构特征和参引特征，即 Web 服务描述文档(WSDL 文档)的两大特征. 根据每个特征对 Web 服务功能语义描述的影响，提出多维向量表示模型. 基于多维向量模型，计算 Web 服务之间的相似度，为服务聚类做准备，提高服务发现效率. 这类聚类方法主要针对 WSDL 或 OWL-S 文档，提取体现服务功能的特征，然后进行聚类. 然而，这类方法的局限性在于文档需要特定的格式，不能用于对用自然语言描述的需求文档中挖掘更有价值的信息.

文献[18]提出了基于主题模型的结构化 Web 服务发现机制，将 Web 服务构成结构化文档，将文档生成词袋，忽略文档中词与词之间的顺序，运用 LDA 算法，获取潜在的文档对应的主题分布. 再者，向量化搜索语句，通过量化的语句计算潜在主题分布，定位到与之相关的 Web 服务，提高服务发现检

索效率. 文献[19]使用 PLSA(Probabilistic Latent Semantic Analysis)和 LDA(Latent Dirichlet Allocation)从服务描述文本中抽取相关主题分布，然后结合 OWL-S 服务的描述和功能属性标签，综合挖取服务潜在的主题分布，再聚类. 文献[20]通过服务描述文本，提取出对应的信息功能集，用 LDA 对提取的功能集进行聚类.

现有的许多服务聚类方法有以下两点考虑不足：(1) 局限于特定的服务文档类型：现有的聚类方法大多局限于 WSDL 文档或 OWL-S(Semantic Markup for Web Service)文档等单一类型的服务描述文档，而较少关注通过自然语言描述的 RESTful 风格类型的服务；(2) 大多数算法并没有考虑语义层面更深一层的考虑：现有算法对于文档利用特征选择或者特征抽取方法，仅保留部分关键特征，必要时通过 LDA 或者 PCA 降维，把数据映射到高维或者低维空间，减少数据计算量. 但是往往计算量大，操作复杂.

利用自然语言处理技术对文档进行特征挖掘在其他领域已经得到广泛研究和应用. 对于服务描述文档而言，这些方法同样适用. 因此，我们在自然语言处理的基础上，提出一种基于服务需求语义的服务聚类方法. 该方法直接从服务描述文本中提取服务功能信息集，不需要生成本体或进行 LDA 降维，适用于多种服务描述文档类型.

5　结论与展望

我们提出了一种基于需求功能语义的服务聚类方法，从服务的需求文档中解析句子语法结构，通过自然语言处理方法提取并处理对应的功能信息集，计算服务间的相似度并运用于 k-means 算法，实现服务聚类. 数据实验使用 PWeb 上 API 服务的真实数据验证了我们方法的有效性. 实验结果表明，我们的方法可以准确地对服务进行聚类，相比其它方法在性能上有所提高

下一步我们将考虑：(1) 我们使用 k-means 算法进行聚类. 然而该算法随机选择中心点，初始中心位置对 k-means 聚类结果有一定的影响. 后续我们将考虑使用性能更好的聚类方法来改进本文的工作；(2) 将本文提出的算法运用于更大的数据集上，用来验证本文工作的有效性.

参　考　文　献

[1] Wu Chao-Hui, Deng Shui-Guang, Wu Jian. Service Computing and Service Technology. Hangzhou: Zhejiang University Press, 2009

[2] Deng Shui-Guang, Yin Jian-Wei, Li Ying, et al. A method of semantic Web service discovery based on bipartite graph matching. Chinese Journal of Computers, 2008, 31(8): 1364-1375(in Chinese)
(邓水光，尹建伟，李莹等. 基于二分图匹配的语义 Web 服务发现方法. 计算机学报，2008，31(8)：1364-1375)

[3] Li Zheng, Wang Jian, Zhang Neng, et al. A topic-oriented clustering approach for domain services. Journal of Computer Research and Development, 2014, 51(2): 408-419(in Chinese)
(李征，王健，张能等. 一种面向主题的领域服务聚类方法. 计算机研究与发展，2014，51(2)：408-419)

[4] Li Zheng, He Ke-Qing, Wang Jian, Zhang Neng. An on-demand services discovery approach based on topic clustering. Journal of Internet Technology, 2014, 15(4): 543-555

[5] Tian Gang, He Ke-Qing, Wang Jian, et al. Domain-oriented and tag-aided Web service clustering method. Acta Electronica Sinica, 2015, 43(7): 1266-1274(in Chinese)
(田刚，何克清，王健等. 面向领域标签辅助的服务聚类方法. 电子学报，2015，43(7)：1266-1274)

[6] Tian Gang, Wang Jian, He Ke-Qing, Sun Cheng-Ai. Leveraging auxiliary knowledge for Web service clustering. Chinese Journal of Electronics, 2016, 25(5): 858-865

[7] Khalid E, Ahmed E H, Patrick M. Clustering WSDL documents to bootstrap the discovery of Web services//Proceedings of the 2010 IEEE International Conference on Web Services. Miami, USA, 2010: 147-154

[8] Liu Yi-Song, Yang Yu-Cheng. Semantic Web service discovery based on text clustering and similarity of concepts. Computer Science, 2013, 40(11): 211-214(in Chinese)
(刘一松，杨玉成. 基于文本聚类和概念相似度的语义 Web 服务发现. 计算机科学，2013，40(11)：211-214)

[9] Pan Wei-Feng, Li Bing, Shao Bo, He Peng. Service classification and recommendation based on software networks. Chinese Journal of Computers, 2011, 34(12): 2355-2369(in Chinese)
(潘伟丰，李兵，邵波，何鹏. 基于软件网络的服务自动分类和推荐方法研究. 计算机学报，2011，34(12)：2355-2369)

[10] Wang Li-Jie, Li Meng, Cai Si-Bo, et al. Internet information search based approach to enriching textual descriptions for public Web services. Journal of Software, 2012, 23(6): 1335-1349(in Chinese)
(王立杰，李萌，蔡斯博等. 基于网络信息搜索的 Web Service 文本描述信息扩充方法. 软件学报，2012，23(5)：1335-1349)

[11] Lin D. An information-theoretic definition of similarity//Proceedings of the 5th International Conference on Machine Learning. USA, 1998: 296-304

[12] Wang Wei-Qiang, Gao Wen. Text mining on the Internet. Computer Science, 2000, 25(4): 32-36(in Chinese)
(王伟强，高文. Internet 上的文本数据挖掘. 计算机科学，2000，25(4)：32-36)

[13] Zhao Y, Karypis G. Criterion functions for document clustering experiments and analysis. Department of Computer Science, University of Minnesota: Technical Report CS0140, 2001

[14] Richi N, Bryan L. Web service discovery with additional semantics and clustering//Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence. Fremont, USA, 2007: 555-558

[15] Christian P, Florian R, Schahram D. Web service clustering using multidimensional angles as proximity measures. Journal of ACM Transactions on Internet Technology, 2009, 9(3): 1-26

[16] Chen Liang, Hu Liu-Kai, Zheng Zi-Bin, et al. WTClustering: Utilizing tags for Web service clustering//Proceedings of the 9th International Conference ICSOC 2011. Paphos, Cyprus, 2011: 204-218

[17] Wei Deng-Ping, Wang Ting, Wang Ji. Web service discovery by integrating structure and reference features of description documents. Journal of Software, 2011, 22(9): 2006-2019

[18] Chen Jiang-Feng, Yu Jian-Jun. Topic model based structural Web services discovery. Journal of Beijing University of Aeronautics and Astronautics, 2008, 34(6): 734-738

[19] Cassar G, Barnaghi P, Moessner K. Probabilistic methods for service clustering//Proceedings of the 2nd European Conference. Málaga, Spain, 2013: 19-33

[20] Zhang Neng, Wang Jian, He Ke-Qing, Li Zheng. An approach of service discovery based on service goal clustering//Proceedings of the 13th IEEE International Conference on Services Computing. San Francisco, USA, 2016: 114-121

JIANG Bo, born in 1970, professor, M. S. supervisor. Her current research interests include service computing, collaborative computing, and software engineering.

YE Ling-Yao, born in 1995, M. S. candidate. His research interest is service computing.

PAN Wei-Feng, born in 1982, Ph. D., associate professor, M. S. supervisor. His current research interests include software engineering, complex networks, and intelligent computation.

WANG Jia-Lei, born in 1991, M. S. candidate. His research interest is service computing.

Background

With the rapid growth of the number and types of services on the Web, how to discover the desired services for users efficiently and accurately has become a significant challenge in service-oriented computing. Service clustering is an important technique to improve the efficiency of service discovery. Although there are many scholars studying the service clustering approaches, the existing approaches are rarely designed from the perspective of the requirement semantics of services. So the focus of the current work is to cluster Web service based on the functional semantics of service requirements. However, it is still a very difficult task to extract functional semantics of requirements and compute the service similarity between every pair of services.

In the current work, a novel service clustering approach, SCFSR (Service Clustering based on Functional Semantics of Requirements), is proposed. First, it uses service requirement text to extract the functional information of services by using natural language processing technologies. Based on functional information, we obtain the semantic similarity between two services and further cluster the services by using k-means algorithm. API services in ProgrammableWeb are used to demonstrate the feasibility of the proposed approach. Experimental results show that the proposed approach can achieve service clustering correctly, which can greatly promote the on-demand service discovery.

This work has been supported by the National Natural Science Foundation of Zhejiang Province under Grant No. LY15F020004, the National Natural Science Foundation of China under Grant No. 61202200, and the Commonweal Project of Science and Technology Department of Zhejiang Province under Grant No. 2015C33091.