Chinese Journal of Electronics
Vol.25, No.2, Mar. 2016

WSGM-SD: An Approach to RESTful Service Discovery Based on Weighted Service Goal Model*

ZHANG Neng[1], HE Keqing[1], WANG Jian[1] and LI Zheng[2]

(1. *State Key Laboratory of Software Engineering, School of Computer, Wuhan University, Wuhan 430072, China*)

(2. *School of Computer and Information Engineering, Henan University, Kaifeng 475001, China*)

Abstract — Faced with the rapidly increasing Web services, it becomes a challenging issue for users to effectively and accurately discover and reuse services. Existing service discovery approaches are mainly developed for services with WSDL documents, while only a few attention is being paid to services described in natural language, especially to the RESTful services. Towards this issue, we introduce a Weighted service goal model (WSGM) by measuring the weights of service goals extracted from the textual descriptions of services. Based on the WSGM, a novel service discovery approach called Service discovery based on WSGM (WSGM-SD) is proposed. Experiments on ProgrammableWeb, a public Web service repository, demonstrate the effectiveness of the proposed approach.

Key words — RESTful service discovery, Weighted service goal model, Text mining.

I. Introduction

Web service discovery aims at finding interested Web services for users, which is the basis of sharing and reusing services. With the rapid growth of services on the Internet, the importance of service discovery is becoming more and more prominent. Currently, there are two mainstream styles of services: SOAP services and RESTful services (or Web APIs). SOAP services should be described using WSDL documents. The heavy-weighted characteristics and over standardization bring much inconvenience to developers. In contrast, RESTful services can be described in both natural language and XML-based languages, *e.g.*, WADL and WSDL 2.0. According to the survey in Ref.[1], many companies often use a simple textual description on Web pages to explain their APIs, instead of using XML-based WADL or WSDL 2.0. By Apr. 14, 2014, nearly 70% of services registered in ProgrammableWeb** (PW) are RESTful services.

Existing service discovery approaches are mainly developed for services with WSDL documents and achieved good results, while only a few attention is paid to the RESTful services described by textual descriptions. Most popular Web service search engines, *e.g.*, PW and Seekda!***, still adopt the keyword-based mechanism for service discovery in essential, which is usually low accuracy and insufficient to satisfy users' high-level goals, *e.g.*, "book a hotel" and "create images". Therefore, the discovery of RESTful services remains an important issue, and the key challenge is to dig out the functionalities from the textual descriptions of services. In our prior work[2], we propose an approach to extract service goals, *i.e.*, domain-specific functionalities, from the services' textual descriptions. However, the approach does not distinguish the importance of different service goals in services, and thus cannot model the functionalities of services accurately. For example, according to the semantics embodied by the name, summary and tags of "BookingSync API" (as shown in Fig.1), the functionalities about "booking" and "rental", *e.g.*, "manage vacation rental bookings", will be more important than those about "client", *e.g.*, "create clients information". Intuitively, measuring the importance of service goals in services will help retrieve better candidate service lists for users.

*Manuscript Received Sept. 25, 2014; Accepted Mar. 24, 2015. This work is supported by the National Basic Research Program of China (No.2014CB340404), the National Natural Science Foundation of China (No.61202031, No.61373037, No.61402150), the Education Department Program of Henan Province (No.14A520008), and the Central Grant Funded Cloud Computing Demonstration Project of China undertaken by Kingdee Software (China).

**http://www.programmableweb.com/*

***http://webservices.seekda.com/*

 DOI:10.1049/cje.2016.03.010

This paper reports our continuous work on the discovery of services with textual descriptions. Firstly, we improve the service goal extraction approach[2] by introducing a Weighted service goal model (WSGM) to capture the importance degrees of service goals in each service. (Note that we use the term "weight" to denote the importance degree of a service goal.) Then, based on the WSGM, we propose a novel service discovery approach WSGM-SD to support the discovery of services with textual descriptions (especially the RESTful services). More specifically, the top k similar services are retrieved by matching the service goals specified in a given user query with those of services, and then reranked by using the weights of matched service goals. Experiment results conducted on real services crawled from PW demonstrate the effectiveness of WSGM-SD.

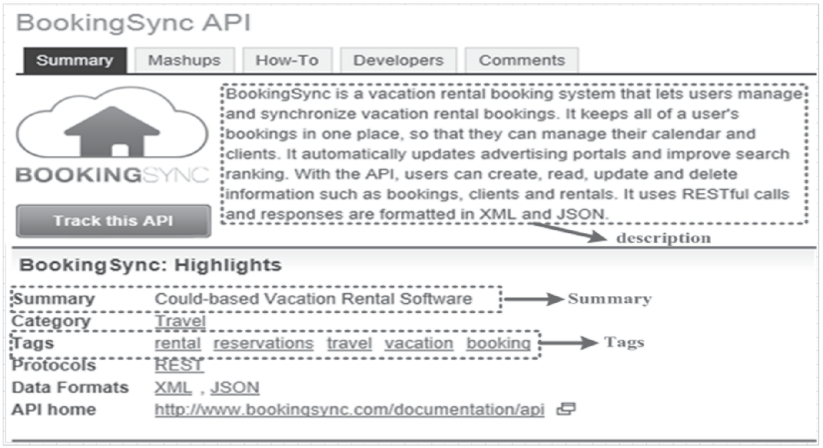


Fig. 1. "BookingSync API" from PW

II. Related Work

Service discovery, a hot topic in Service-oriented computing (SOC), has been widely investigated in the past decades. For example, Plebani *et al.*[3] propose an approach URBE for service retrieval by evaluating the similarity between service interfaces. Klusch *et al.*[4] present a hybrid semantic matchmaker for SAWSDL, which considers both semantic similarity and structural similarity between services. Moreover, several researches leverage bipartite graph to calculate similarity between services based on WSDL documents[5,6].

In recent years, some machine learning techniques have also been applied in service discovery. More specifically, Elgazzar *et al.*[7] propose a technique to extract features from WSDL document and then cluster services into functionally similar groups. Chen *et al.*[8] propose a service clustering approach WT-LDA, which integrates service tagging data and WSDL document through augmented Latent dirichlet allocation (LDA). These methods can help to improve the time efficiency of service discovery. While many approaches (as those mentioned above) proposed for discovering services with WSDL documents, only a few works reported the discovery of services described using short text, especially the RESTful services. Most RESTful service discovery approaches are based on hRESTS[9], a machine-readable microformat for RESTful services, which is built on the HTML service documentation whose target audience are developers. There are also some RESTful service discovery works[10−12] that are based on MicroWSMO (a semantic extension of hRESTS). The major limitation of these works is that the HTML service documentation including operational description used for hRESTS construction is unavailable or hard to be obtained in most public web service portals such as Seekda! and PW.

Another kind of related work is about mining functionality from text, which is widely investigated in other domains[13,14]. For example, Ghose *et al.*[13] realize a toolkit R-BPD which uses the syntax parser NLTk to identify business processes. Friedrich *et al.*[14] present an approach to generate BPMN models from text using natural language tools, *e.g.*, the Stanford parser, FrameNet and WordNet. In contrast, we adopt a different strategy to extract service goals from the textual descriptions of services based on the ranked keyword lists of domains.

III. Service Discovery Framework

As shown in Fig.2, our service discovery framework consists of three major components: 1) domain-oriented service categorization, 2) construction of WSGM, and 3) service discovery. In the first component, the services will be categorized into different domains according to our prior work[15]. A ranked keyword list of each domain (denoted as DRKL) will also be produced. In the second component, we extract the service goals contained in the textual descriptions of services. Afterwards, the WSGM of each service will be constructed by measuring the weights of service goals based on some useful infor-

mation, *e.g.*, service name and user tagging data. Since these two components can be done offline, the efficiency of service discovery is not a big concern, while the accuracy is more important. In the third component, the WSGM will be used to improve the performance of service discovery. More specifically, when a user query is sent to the Web service search engine, the target domain to which the query belongs will be identified first, and then service goals contained in the query will be extracted. Afterwards, the top k similar services can be obtained by matching the service goals extracted from the query with those of services, and then reranked by using the weights of matched service goals to return a better service list.

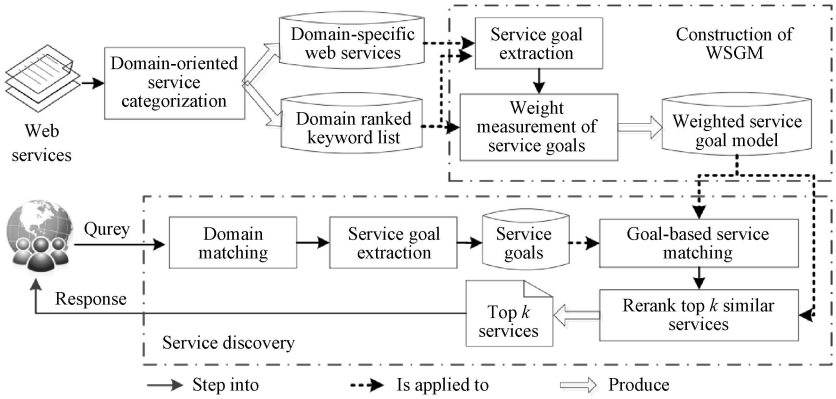


Fig. 2. Service discovery framework

The service goal is used to exhibit the intentional functionality of a service. In Ref.[16], according to some related works on goal modeling[17,18] and the fact that the naming of WSDL operations always adopts a verb-noun form, *e.g.*, "GetWeatherByCityState", we define the service goal as follows:

Definition 1 A service goal sg is defined as a triple $\langle sgv, sgn, sgp\rangle$, where sgv is a verb or verb phrase, which denotes the action of the service goal, sgn is a noun or noun phrase, which denotes the entities affected by the action, and sgp is a set of parameters, which denotes some additional information such as the initial or final state of the entity, and how the action affects the entity. Note that for a specific sg, sgv and sgn are mandatory, while sgp is optional.

IV. Construction of WSGM

In this section, we describe the construction of WSGM, which consists of two phases: service goal extraction and weight measurement of service goals.

1. Service goal extraction

As shown in Fig.1, the textual description of a service usually contains several sentences. In Ref.[2], we propose an approach to extract service goals from the textual description based on the dependencies of sentences generated by the Stanford Parser*, which can reflect the grammatical relations between the words in a sentence. Fig.3 depicts the dependencies of the forth sentence, denoted as $ESen4$, in the description of "BookingSync API". Note that the sentences contained in the textual description of a service can be identified using the document preprocessing tool included in the Stanford parser.

The process of the service goal extraction approach comprises three steps, which are briefly described below.

1) Initial goal generation

By analyzing various kinds of sentences, we find that a sentence may contain several service goals, and some of them can be easily extracted from the dependencies while others are hard to be extracted. For example, given the dependencies in Fig.3, a simple service goal ⟨create, information, null⟩ can be extracted directly from the dependency dobj(create-7, information-14), while others cannot, *e.g.*, ⟨update, rentals, null⟩. Based on the sentence analysis, we summarized three dependency patterns: 1) $nsubjpass(a, b) \rightarrow \langle a, b, \text{null}\rangle$, 2) $dobj(a, b) \rightarrow \langle a, b, \text{null}\rangle$, and 3) $prep(a, b)$ & $aux(a, c) \rightarrow \langle a, b, \text{null}\rangle$, to extract these easily extracted service goals (named as initial goals) of sentence Sen, denoted as $IG(Sen)$, *e.g.*, $IG(ESen4) = \{\langle \text{create, information, null}\rangle\}$.

2) Initial goal extension

As shown in $IG(ESen4)$, the initial goals of a sentence will not be the desired service goals due to two issues. On one hand, the sgv and sgn parts of some initial goals may lose some meaningful information, and the sgp parts of some initial goals need to be filled. On the other hand, there are some potential service goals need to be supplemented. To solve these issues, the initial goals need to be extended by considering more de-

*http://nlp.stanford.edu/software/lex-parser.shtml

pendencies, as shown in Table 1. Based on the extension results of initial goals, the candidate service goals of sentence Sen, denoted as $CSG(Sen)$, can be obtained, *e.g.*, $CSG(ESen4)$ = {⟨create, information, sgp⟩, ⟨read, information, sgp⟩, ⟨update, information, sgp⟩, ⟨delete, information, sgp⟩} (sgp = {with API, such_as bookings, such_as clients, such_as rentals}).

Finally, the candidate service goals of service S, denoted as $CSG(S)$, can be obtained by: $CSG(S) = \bigcup_{Sen_i \in SEN(S)} CSG(Sen_i)$, where $SEN(S)$ denotes all sentences contained in the textual description of S.

1. det(API-3, the-2)	7. conj_and(create-7, update-11)
2. prep_with(create-7, API-3)	8. conj_and(create-7, delete-13)
3. nusubj(create-7, users-5)	9. dobj(create-7, information-14)
4. aux(create-7, can-6)	10. prep_such_as(information-14,bookings-17)
5. root(ROOT-0, create-7)	11. conj_and(bookings-17, clients-19)
6. conj_and(create-7, read-9)	12.conj_and(bookings-17, rentals-21)

Fig. 3. Dependencies of $ESen4$ in "BookingSync API"

Table 1. Dependencies for initial goal extension

Dependency	Usage	Example phrase (EP)	Relevant dependencies in EP
prt	Identify the particle of sgv	look up	prt(look-1, up-2)
conj	Identify coordinate verb(s) of sgv or coordinate noun(s) of sgn	1) create, read, update and delete 2) bookings, clients and rentals	As shown in Fig. 3
prep	Identify prepositional phrase(s) of sgv or sgn	1) information of bookings 2) search hotels by city	1) $prep\_of$(information-1, bookings-3) 2) $prep\_by$(search-1, city-4)
appos	Identify appositive noun(s) of sgn, which can also be seen as coordinate noun(s) of sgn	information such as bookings, clients	$appos$(bookings-4, clients-6)
poss	Identify the possessive determiner or genitive's complement of sgn	1) tag's recommendation 2) your data	1) $poss$(recommendation-3, tag-1) 2) $poss$(data-2, your-1)
amod	Identify adjectival modifier(s) of sgn	newest articles	$amod$(articles-2, newest-1)
nn	Identify noun modifier(s) of sgn	vacation rental bookings	nn(bookings-4, vacation-2); nn(bookings-4, rental-3);

3) Service goal refinement

The candidate service goals may have some insufficiencies. Firstly, they may contain domain irrelevant terms, *e.g.*, "let", "more" and "API". Secondly, there are some words with the same stem, *e.g.*, "retrieve", "retrieved" and "retrieving" have the same stem "retrieve". Thirdly, the sgn parts of some candidate service goals may contain only abstract nouns, *e.g.*, "information" and "list". To solve these problems, the candidate service goals are refined by several steps including stemming, stop word removal, and supplement of sgn. In particular, in the supplement of sgn, the sgn parts of two special kinds of candidate service goals: 1) sgn == null and 2) sgn only contains abstract nouns, will be supplemented with important nouns contained in the sgp parts. Note that the important nouns can be those in top 50 of the DRKL. For example, ⟨create, information, {such_as booking, such_as client, such_as rental}⟩ $\in CSG(ESen4)$ will be divided into: {⟨create, booking information, null⟩, ⟨create, client information, null⟩, ⟨create, rental information, null⟩}.

2. Weight measurement of service goals

Generally, a service may contain several service goals and some of them may be more important than others. For example, in "BookingSync API", the service goals about "booking", *e.g.*, ⟨create, booking information, null⟩, will be more important than those about "client", *e.g.*, ⟨create, client information, null⟩, which can be judged by the name, summary and tags of "BookingSync API". Moreover, the importance degrees (*i.e.*, weights) of service goals can also be reflected by: 1) the frequencies of words in the service's description (*e.g.*, the frequencies of "booking" and "client" are 4 and 2, respectively), and 2) the rankings of words in the DRKL (*e.g.*, the rankings of "booking" and "client" are 2 and 13 in the DRKL, respectively). Intuitively, the weights of service goals can help model the functionalities of a service more accurately, which can further help retrieve better service lists for users. In this paper, we attempt to measure the importance degrees of service goals in a service based on the importance degrees of constituent words of service goals by considering the following factors. Note that the value ranges of all the factors will be normalized to $[0, 1]$ in order to ensure that the weighted sum of importance degrees of words and service goals will be restricted to $[0, 1]$.

1) Service name relevance (SNR)

The service name assigned by the service provider is intended to reflect the central topic of a service. The words relevant to the service name will be important. As "BookingSync API" shows that the service name may be a combination of several words and some words may be abbreviated, we adopt the longest common string metric to compute the SNR of word w in service S:

$$SNR_{S,w} = \frac{len(comStr(SN, w))}{len(w)} \tag{1}$$

where SN denotes the name of S, $comStr(SN, w)$ is to obtain the longest common string between SN and w, *e.g.*, $comStr$("BookingSync API", "booking") = "booking". $len(*)$ is the length of string $*$.

2) Summary relevance (SMR)

In a service, the summary provided by the service provider is a short summarized description of the service's main functionalities. Any word contained in the summary will be important for the service. Thus, we define the SMR of word w in service S as follows:

$$SMR_{S,w} = \begin{cases} 1, & w \in SUM_S \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where SUM_S denotes the words in the summary of S, which need to be preprocessed by stemming and stop word removal.

3) Tag relevance (TR)

The tags of a service are keywords assigned by different users (*e.g.*, service provider and consumers) which can reflect the key functionalities of the service. The words contained in tags are also important for the service. The TR of word w in service S can be defined by:

$$TR_{S,w} = \begin{cases} 1, & w \in TAG_S \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where TAG_S denotes the tags of S, which need to be stemmed.

4) Service representation (SR)

Term frequency-inverse document frequency (TF-IDF) is a widely adopted metric to reflect how important a word is to a document. According to TF-IDF, in a document, the words (except the very frequent words appeared in most documents, *i.e.*, stop words) with higher frequency will be more important. Since stop words have been removed from service goals, we simply use the term frequency to define the SR of word w in service S:

$$SR_{S,w} = \frac{\log_2 f_{S,w}}{\max_{w_j \in SGW_S}\{\log_2 f_{S,w_j}\}} \tag{4}$$

where $f_{S,w}$ denotes the frequency of w in S and the purpose of logarithm operation is to minimize the effect of the frequency. SGW_S denotes all words contained in the service goals of S. The other words do not need to be considered since they are not useful for measuring the weights of service goals.

5) Domain representation (DR)

As stated previously, the rankings of words in the DRKL can reflect the weights of service goals from a domain perspective. Specifically, the service goals with higher ranked words in DRKL will be more important. As the $kf-irf$ values of words in DRKL can reflect the weights of words more accurately than the rankings, we define the DR of word w in domain d as follows:

$$DR_{d,w} = \frac{kf-irf_{d,w}}{\max_{w_j \in SGW_S}\{kf-irf_{d,w_j}\}} \tag{5}$$

where $kf-irf_{d,w}$ denotes the $kf-irf$ value of w in d.

Based on the above factors, the weight of word w in service S can be computed by:

$$\begin{aligned} \mathcal{W}_{S,w} = {} & w_1 \times SNR_{S,w} + w_2 \times SMR_{S,w} + w_3 \times TR_{S,w} \\ & + w_4 \times SR_{S,w} + w_5 \times DR_{d,w} \end{aligned} \tag{6}$$

where $w_1 \sim w_5$ are weight coefficients which can be adjusted by users with the restriction of $\sum_{i=1}^{5} w_i = 1$. According to their origins, the factors can be divided into two groups: 1) human assigned factors, *i.e.*, SNR, SMR and TR, and 2) statistics-based factors, *i.e.*, SR and DR. Since the human assigned factors involve human intelligence, the priority of them should be higher than the statistic-based factors. More specifically, as explained above, the priority between each of these factors can be: $w_1 \geq w_2 \geq w_3 > w_4 \geq w_5$.

Then, the weight of service goal sg in service S can be calculated as follows:

$$\begin{aligned} \mathcal{W}_{S,sg} = {} & \alpha \times \sum_{w_i \in W(sg.sgv)} \mathcal{W}_{S,w_i} + \beta \times \sum_{w_j \in W(sg.sgn)} \mathcal{W}_{S,w_j} \\ & + \gamma \times \sum_{w_k \in W(sg.sgp)} \mathcal{W}_{S,w_k} \end{aligned} \tag{7}$$

where $W(part)$ denotes the words except prepositions contained in the phrase *part*. Since *sgv* and *sgn* are mandatory for service goals while *sgp* is optional, and *sgn* is more useful to reflect the users' searching intention, the priority among them should be: $sgn \geq sgv > sgp$. Therefore, we set: $\beta \geq \alpha > \gamma$.

Finally, the weighted service goal model of service S can be defined as: $WSGM_S = \langle SG(S), \mathcal{WM} \rangle$, where $SG(S)$ denotes the service goals of S, $\mathcal{WM} = \{\langle sg_i, \mathcal{W}_{S,sg_i} \rangle | sg_i \in SG(S)\}$ stores the weights of service goals.

V. Service Discovery

After the WSGM of all services have been constructed, the process of service discovery in Fig.2 will be initiated when a user query, denoted as Q, is sent to the Web service search engine. Firstly, Q will be preprocessed including word segmentation, stemming, stop word removal and word frequency count. Then, the similarity between Q and each domain in the service registry can be calculated using Eq.(3) in Ref.[16], and the most similar domain can be chosen as the target domain (denoted by td) to which Q belongs. Afterwards, the service goals contained in Q, denoted as $SG(Q)$, can be extracted using the service goal extraction approach.

1. Goal-based service matching

Service matching, the central task of service discovery, is to calculate the similarities between the query Q and services. In this paper, the functional similarity between Q and service S in td can be calculated by matching the

service goals of Q with those of S as follows:

$$Sim(Q,S)=\frac{\sum_{sg_i\in SG(Q)}\max_{sg_j\in SG(S)}\{gSim(sg_i,sg_j)\}}{SG(Q)} \tag{8}$$

where $gSim(sg_i,sg_j)$ is the similarity between sg_i and sg_j, which can be calculated by:

$$\begin{aligned}gSim(sg_i,sg_j)=&\alpha\times WSim(sg_i.sgv,sg_j.sgv)\\&+\beta\times WSim(sg_i.sgn,sg_j.sgn)\\&+\gamma\times WSim(sg_i.sgp,sg_j.sgp)\end{aligned} \tag{9}$$

where α,β,γ are weight coefficients referring to Eq.(7). $WSim(A,B)$ denotes the semantic similarity between the words in phrases A and B, *i.e.*, $W(A)$ and $W(B)$, which can be calculated as follows:

$$WSim(A,B)=\frac{\sum_{w_i\in W(A)}\max_{w_j\in W(B)}\{wsim(w_i,w_j)\}}{W(A)} \tag{10}$$

where $wsim(w_i,w_j)$ denotes the semantic similarity between w_i and w_j in WordNet*.

According to the similarity calculated by Eq.(8), top k similar service list of Q, denoted as $SSL_k(Q)$, can be obtained.

2. Rerank top k similar services

Since the top k similar services in $SSL_k(Q)$ are only ranked by similarity, services with the same similarity will remain disordered. To return a better service list, the services with the same similarity in $SSL_k(Q)$ can be ranked by using the weights of service goals. Firstly, $SSL_k(Q)$ will be divided into several segments according to the similarity: the similarities of services in one segment are the same, while the similarities of services in different segments are not. Then, the services in each segment can be ranked by two steps: 1) For each service S, we obtain the matched service goals that satisfy Q, denoted by $SG(S/Q)$, as follows:

$$\begin{aligned}SG(S/Q)=\{sg_i|&sg_i\in SG(S)\wedge\exists sg_j\forall sg_k\{sg_j\in SG(Q)\\&\wedge sg_k\in SG(S)\wedge k\neq i\rightarrow\\&gSim(sg_j,sg_i)\geq gSim(sg_j,sg_k)\}\}\end{aligned} \tag{11}$$

It can be seen that each service goal in $SG(S/Q)$ should have the highest similarity with a certain service goal in $SG(Q)$. Note that a threshold Th can be set to filter the service goals whose highest similarities are less than Th.

2) Rank services in descending order according to the weights of service goals in $SG(S/Q)$, *i.e.*, $\sum_{sg_i\in SG(S/Q)}\mathcal{W}_{S,sg_i}$.

VI. Experiments

In this section, we conduct experiments to evaluate the service goal extraction approach (denoted as SGE) and WSGM-SD. All experiments are conducted on a PC with 2GHz Intel Core i5 CPU and 4GB RAM, running Windows 7 OS.

1. Dataset and metrics

We collected 11,282 services from PW as our testbed (on April 14, 2014). For each service, we get the data of service name, tags, summary, description and category. As the manual creation of ground truth is an expensive process, we randomly select three large domains: Photos, Enterprise and Payment, for experiment. The number of services in Photos, Enterprise and Payment are 254, 491 and 325, respectively. As preparation, we classified the collected services for each selected domain according to the service categorization approach[15]. Afterwards, the results of SGE and service discovery approaches will be evaluated using the Precision and Recall metrics, which are widely adopted in information retrieval.

$$\text{Precision}=\frac{|R_A\cap R_M|}{R_A},\quad \text{Recall}=\frac{|R_A\cap R_M|}{R_M} \tag{12}$$

where R_A denotes the set of automatically extracted service goals or the set of services returned by service discovery approaches; R_M denotes the set of manually extracted service goals or the set of manually identified services that satisfy a specific query.

2. Performance of service goal extraction

To evaluate the performance of SGE, we randomly selected 50 services from each selected domain (150 services in total). Then, we recruited three master students to manually extract service goals from the selected services, and compare their extracted service goals with those extracted using SGE. Table 2 shows the Average (AVG) evaluation results of three domains based on the evaluation results of three students on each service in the domains. It can be seen that the AVG Precision and AVG Recall of three domains are all around 85%, which indicates that our approach can extract service goals from the textual descriptions of services effectively.

Table 2. Evaluation result of SGE

	AVG precision	AVG recall
Photos	0.8989	0.8658
Enterprise	0.8495	0.8536
Payment	0.8441	0.8446

3. Performance of service discovery

In this section, we compare the performances of four service discovery approaches, which are detailed as follows:

Service discovery based on Keyword matching (KWMSD): In this approach, the similar services are discovered by matching the keywords of services with the keywords contained in the query.

*http://wordnet.princeton.edu/

Service discovery provided in PW (PWSD): This approach refers to the service searching mechanism provided in PW.

Service discovery based on Weighted service goal model (WSGM-SD): In this approach, the top k similar services are retrieved and reranked by using the WSGM, as detailed in Section V. According to the weight coefficients discussed in Eqs.(6) and (7), we have the following settings: $w_1 = 0.3$, $w_2 = 0.25$, $w_3 = 0.2$, $w_4 = 0.15$, $w_5 = 0.1$, and $\alpha = 0.35$, $\beta = 0.5$, $\gamma = 0.15$.

Service discovery based on service goal model (SGM-SD): This approach is a simplified version of WSGM-SD without reranking the top k similar services.

To evaluate the performances of service discovery approaches, we performed the service discovery approaches on several queries: 1) "create images" (Query-1), 2) "manage contact information" (Query-2), and 3) "I need to process transactions" (Query-3). Table 3 shows the time cost of service discovery on the queries. Since the resource platform of PW is unknown, the comparable time cost of PWSD cannot be obtained, denoted as n/a. We can see that the time costs of WSGM-SD (or SGM-SD) are only a little higher than KWMSD, which demonstrate the efficiency of WSGM-SD.

Table 3. Time cost of service discovery on queries

Query	Time cost of service discovery			
	KWMSD	PWSD	SGM-SD	WSGM-SD
Query-1	234ms	n/a	283ms	285ms
Query-2	231ms	n/a	298ms	299ms
Query-3	238ms	n/a	314ms	316ms

As users are usually interested in several top services in the resulting service list, the evaluation will focus on the top k services returned by service discovery approaches, and the R_M in Eq.(12) is simplified as $R_M = \bigcup_{SD_i \in SDS} R_M(SD_i@k)$, where $SDS = \{\text{KWMSD, PWSD, SGM-SD, WSGM-SD}\}$, $R_M(SD_i@k)$ denotes the set of services that can satisfy the query contained in the top k services returned by SD_i. Table 4 shows the evaluation results of top 10 and top 20 services. We can see that WSGM-SD (or SGM-SD) outperforms the other two approaches in terms of both Precision and Recall. It can also be found that the Precision and Recall values of SGM-SD and WSGM-SD are the same in most cases. However, it cannot be claimed that the performances of SGM-SD and WSGM-SD are equivalent in most cases since both Precision and Recall do not consider the rankings of the resulting services. To evaluate the performances of service discovery approaches more precisely, we adopt the Discounted cumulative gain (DCG), a widely adopted metric in Web search, which is defined as follows:

$$DCG_k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{13}$$

where i is the ranking position in the service list and rel_i is the relevance of the ith service. The higher the DCG value of the resulting service list is, the better the service discovery approach will be.

Table 4. Precision and recall of services discovery approaches

	Query-1		Query-2		Query-3	
	Precision	Recall	Precision	Recall	Precision	Recall
Top 10						
KWMSD	0.7	0.35	0.9	0.3913	0.7	0.3182
PWSD	0.7	0.35	0.7	0.3044	0.6	0.2727
SGM-SD	1.0	0.5	1.0	0.4348	1.0	0.4546
WSGM-SD	1.0	0.5	1.0	0.4348	1.0	0.4546
Top 20						
KWMSD	0.65	0.3939	0.75	0.5172	0.6	0.3429
PWSD	0.7	0.4242	0.75	0.5172	0.65	0.3714
SGM-SD	0.9	0.5455	0.9	0.6207	0.95	0.5429
WSGM-SD	1.0	0.6061	0.9	0.6207	0.95	0.5429

Fig.4 shows the DCG values of top 5 and top 10 services returned by service discovery approaches. We can see that WSGM-SD is better than the other three approaches including SGM-SD, which denotes that the service lists returned by WSGM-SD are more desired by users, and that reranking services by using the weights of service goals can contribute to better service lists.

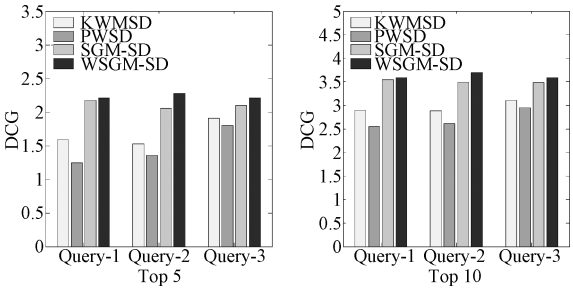


Fig. 4. DCG of top 5 and top 10 services

Moreover, we conducted experiments to investigate the impact of different settings of the weight coefficients in Eq.(6), which are significant for measuring the weights of service goals. As stated previously, the priority of those weight coefficients can be: $w_1 \geq w_2 \geq w_3 > w_4 \geq w_5$. To validate this priority order, we tested three representative settings: 1) Setting-1: $w_1 \sim w_5$ are set as 0.3, 0.25, 0.2, 0.15, and 0.1, respectively; 2) Setting-2: $w_1 \sim w_5$ are set equally, that is, each is set to be 0.2; 3) Setting-3: $w_1 \sim w_5$ are set as 0.1, 0.15, 0.2, 0.25, and 0.3, respectively. The DCG values of the service lists returned by WSGM-SD with these settings are shown in Table 5. We can see that the DCG values of Setting-1 are a little better than the other Settings.

VII. Conclusion

In this paper, we report our recent work on the discovery for services with textual descriptions, especially

the RESTful services. Firstly, we extract the service goals from the textual descriptions of services using the approach proposed in Ref.[2], and then build a weighted service goal model for each service by measuring the weights of service goals according to some semantic information. Afterwards, a novel service discovery approach WSGM-SD is proposed. Experiments conducted on real services crawled from PW demonstrate the effectiveness of the proposed approach.

Table 5. DCG of WSGM-SD with different settings

	Query-1	Query-2	Query-3
	Top 5		
Setting-1	2.2120	2.2794	2.2120
Setting-2	2.2120	2.2733	2.2084
Setting-3	2.2042	2.2697	2.2046
	Top 10		
Setting-1	3.5830	3.6944	3.5868
Setting-2	3.5830	3.6935	3.5794
Setting-3	3.5752	3.6884	3.5756

References

[1] W. Jiang, D. Lee and S. Hu, "Large-scale longitudinal analysis of SOAP-based and RESTful web services", *Proc. of IEEE International Conference on Web Services*, Honolulu, Hawaii, USA, pp.218–225, 2012.

[2] N. Zhang, J. Wang, Z. Li, *et al.*, "Approach of constructing domain service goal knowledgebase", *Journal of Chinese Computer Systems*, Vol.35, No.9, pp.1943–1948, 2014.

[3] P. Plebani and B. Pernici, "URBE: Web service retrieval based on similarity evaluation", *IEEE Transactions on Knowledge and Data Engineering*, Vol.21, No.11, pp.1629–1642, 2009.

[4] M. Klusch, P. Kapahnke and I. Zinnikus, "Hybrid adaptive web service selection with SAWSDL-MX and WSDL-analyzer", *Proc. of European Semantic Web Conference*, Heraklion, Crete, Greece, pp.550–564, 2009.

[5] J. Zhang, R. Madduri, W. Tan, *et al.*, "Toward semantics empowered biomedical web services", *Proc. of IEEE International Conference on Web Service*, Washington, DC, USA, pp.371–378, 2011.

[6] F. Liu, Y. Shi and J. Yu, "Measuring similarity of web services based on WSDL", *Proc. of IEEE International Conference on Web Services*, Miami, Florida, USA, pp.155–162, 2010.

[7] K. Elgazzar, A.E. Hassan and P. Martin, "Clustering wsdl documents to bootstrap the discovery of web services", *Proc. of International Conference on Web Services*, Los Angeles, CA, USA, pp.147–154, 2009.

[8] L. Chen, Y. Wang, Q. Yu, *et al.*, "WT-LDA: User tagging augmented LDA for web service clustering", *Proc. of International Conference on Service-Oriented Computing*, Berlin, Germany, pp.162–176, 2013.

[9] K. Jacek, G. Karthik and V. Tomas, "hRESTS: An HTML microformat for describing RESTful web services", *Proc. of International Conference on Web Intelligence and Intelligent Agent Technology*, Sydney, NSW, Australia, pp.619–625, 2008.

[10] U. Lampe, S. Schulte, M. Siebenhaar, *et al.*, "Adaptive matchmaking for RESTful services based on hRESTS and MicroWSMO", *Proc. of the 5th International Workshop on Enhanced Web Service Technologies*, Ayia Napa, Cyprus, pp.10–17, 2010.

[11] F. Slaimi, S. Sellami, O. Boucelma, *et al.*, "Flexible matchmaking for RESTful web services", *OTM Conferences, LNCS*, Springer, pp.542–554, 2013.

[12] D. Roman, J. Kopecky, T. Vitvar, *et al.*, "WSMO-Lite and hRESTS: Lightweight semantic annotations for web services and RESTful APIs", *Journal of Web Semantics*, Vol.31, pp.39–58, March 2015.

[13] A. Ghose, G. Koliadis and A. Chueng, "Rapid business process discovery (R-BPD)", *Proc. of International Conference on Conceptual Modeling*, Auckland, New Zealand, pp.391–406, 2007.

[14] F. Friedrich, J. Mendling and F. Puhlmann, "Process model generation from natural language text", *Proc. of CAiSE*, London, UK, pp.482–496, 2011.

[15] J. Zhang, J. Wang, P.C.K. Hung, *et al.*, "Leveraging incrementally enriched domain knowledge to enhance service categorization", *International Journal of Web Services Research*, Vol.9, No.3, pp.43–66, 2012.

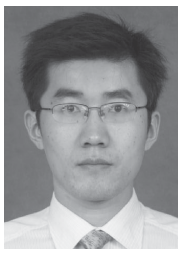
[16] J. Wang, N. Zhang, C. Zeng, *et al.*, "Towards services discovery based on service goal extraction and recommendation", *Proc. of International Conference on Services Computing*, Santa Clara Marriott, CA, USA, pp.65–72, 2013.

[17] C. Rolland, C. Souveyet and C.B. Achour, "Guiding goal modeling using scenarios", *IEEE Transactions on Software Engineering*, Vol.24, No.12, pp.1055–1071, 1998.

[18] M. Strohmaier, M. Lux, M. Granitzer, *et al.*, "How do users express goals on the web? – An exploration of intentional structures in web search", *Proc. of International Conference on Web Information Systems Engineering*, Nancy, France, pp.67–78, 2007.

ZHANG Neng was born in 1990. He is a Ph.D. candidate at Computer School, Wuhan University. His research interests include software engineering, service computing and knowledge acquisition. (Email: nengzhang@whu.edu.cn)

HE Keqing (corresponding author) was born in 1947. He is a professor and Ph.D. supervisor at Wuhan University. His research interests include service computing and software engineering. (Email: hekeqing@whu.edu.cn)



WANG Jian was born in 1980. He is a Ph.D., lecturer at Wuhan University, and a member of China Computer Federation (CCF). His current research interests include requirement engineering and service computing. (Email: jianwang@whu.edu.cn)

LI Zheng was born in 1984. She is a Ph.D., lecturer at School of Computer and Information Engineering, Henan University. Her research interests include service computing and software engineering. (Email: zhengli_hope@whu.edu.cn)