A Web Service Clustering Method with Semantic Enhancement Based on RGPS and BTM

Fang Xie[1], *Jing-Liang Chen*[1*], *Yi Zhu*[2], *Hong-Yan Zheng*[3]

[1] *School of Computer Science, Hubei University of Technology, China*
[2] *School of Transportation & Information, Hubei Communications Technical College, China*
[3] *College of Information and Communication, National University of Defense Technology, China*
thanks_xf@hotmail.com, chen@hbut.edu.cn, zhuyi22250@163.com, doriswinner@163.com

Abstract

In order to overcome the data sparsity problem in service description text and to improve the web service clustering quality, we propose a web service clustering method with semantic enhancement based on RGPS (Role-Goal-Process-Service) Framework and Bi-term Topic Model (BTM). First, we extend service description text's feature according to RGPS meta-model framework. Also, we generate the service latent feature by BTM. Then, we employ K-means on the generated features. The results of experiments on service registry PWeb show that this method can get better clustering performance in purity and entropy. It is proved that this method has great efficiency compared with the baseline methods K-means, Agglomerative and LDA (Latent Dirichlet Allocation). This paper enhances the service clustering performance and creates foundation work for service organization and recommendation.

Keywords: Web service clustering, RGPS meta-model, BTM, K-means

1 Introduction

The web services are growing very rapidly [1]. Web service clustering plays the important role in service organization, management, discovery and recommendation [2-4]. Since service clustering can be used to discover the hidden structure which is revealed by the amount of data points [5]. Tang et al. [6] proposed a general way to organize service into feature vectors according to its WSDL file, which is effective in many applications. Service clustering and classification are highly recommended for using in intrusion detection. The task of service classification is to determine the web service category. When we want to search a new web service in the repository, it should select a category to which the web service belongs depending on the platform's classification. The service clustering task is aim to vectorize the web service depending on the documents and create the clusters according to their similarity. Generally, clustering consists of many sub-processes: for example, preprocess, feature selection, clustering algorithm and evaluation [7-8]. When the web service description text is sparse, similarity calculation accuracy is usually limited [9]. According to the data from web service registry PWeb, the service description document is a short text, and it is described in natural language. Through statistical analysis of the service description text, the average number of words is almost 72 [10]. How to perform service clustering effectively in the context of semantic sparsity has become a big challenge.

As an essential topic, many methods have been put forward for the above problem. Traditional topic model methods like LDA [9] and PLSI [11] represent texts as a mixture of topics and use the text word co-occurrence to search text topics. BTM [12] extends it to a more principal approach by modeling the generative process of word co-occurrence patterns in a corpus, which avoids the data sparisity. Among the relevant text mining tools are document classification, sentiment analysis and topic modeling. Topic modeling refers to find the semantic structure in the texts. Topic modeling assumes that the latent topics of text must exist in the text corpus, and topics should be revealed through the statistical approaches. BTM is very similar to LDA, which both of them are topic modeling method, but it has the more advantages in working with the short text. It can use the co-occurrence information of unordered pairs of words to overcome the sparsity problem, depending on the text corpus to reveal the topics. However, it is an extraordinary task for the two major challenges in BTM. First, topic extraction, for the sparsity text, the existing algorithm dealing with the long text can not deal it with well. Second, most traditional text semantic processing models have not considered context semantics, which may lead to inaccurate results.

Many approaches usually characterize the sparsity text by semantic association or using the external knowledge database [13]. For example, Wikipedia is used in [14] as the external knowledge to extend the text content. Phan et al. [15] proposed a topic for multi-granularity, and a discriminative feature is generated for sparse data clustering. Cataldi et al. [16] uses semantic relationship rule to construct a rule library to enrich the feature corpus.

To solve the above problem, this paper focuses on semantic sparsity by applying RGPS [17] framework as a versatile meta-model structure to extend the semantic content of service. The web service's feature is organized according to the corresponding models and the dependent relationships among RGPS elements. The service feature is extended from four aspects. The extended feature can better describe the service's hidden structure. BTM is used for generating

*Corresponding Author: Jing-Liang Chen; E-mail: chen@hbut.edu.cn

DOI: 10.53106/160792642023072404012

the latent feature of service so that we can utilize topics information in K-means algorithm [18]. Then the K-means is acted on it to obtain the service clustering result. The K value of the algorithm is adaptive, and clustering is incremental.

In this paper, there are the two main contributions:

(1) We introduce RGPS as a feature extension meta-model framework that expands service features based on its description document. Our implementation of service feature extension and dataset used in our experiments are publicly available.

(2) The service topic clustering (BTM+K-means) performs better than other baseline clustering methods.

Our paper is organized as follows. The proposed method is reviewed in Section 2. Section 3 introduces the service feature extension based on RGPS meta-model framework. Section 4 discusses our method. Section 5 details the implementation and evaluation. Section 6 concludes our work.

2 Related Works

Since clustering is used to discover the hidden structure from a dataset which is revealed by many data features, many clustering methods have been proposed for a lot of applications.

Service clustering is the foundation of service organization and recommendation. Method for service recommendation based on the auxiliary knowledge involves collecting and handling web service description information and obtaining web service description keywords and clusters.

2.1 Research on Text Clustering

Wang et al. [19] proposed a text clustering approach using the technology of word embedding and neural network. It learns the representation of text using a convolutional neural network on which clustering is performed. Hadifar et al. [20] proposed a text clustering approach using the technology of weighted word embedding.

LDA is a unsupervised topic model. Many approaches have been proposed to deal with sparsity problem in the short text. Zhang et al. [21] adopts an improvement LDA model which takes the relationship between user information and text relevance, and the model is expanded by means of users and links. Xie et al. [22] proposes an RT-LDA model to resolve the word limit. It adopts Gibbs sampling to mine the topic of every piece of post, and maps the text to dimensions. Chen et al. [23] proposed to extend the features of WSDL document and web service tags based on the function, then they incorporated the tags. In our paper, we present RGPS meta-model framework to extend service feature that is similar to this kind of method. The related feature words are added to enhance the topic characteristics of the service description document.

2.2 Service Clustering

In this paper, we use the vector to represent the service description text.

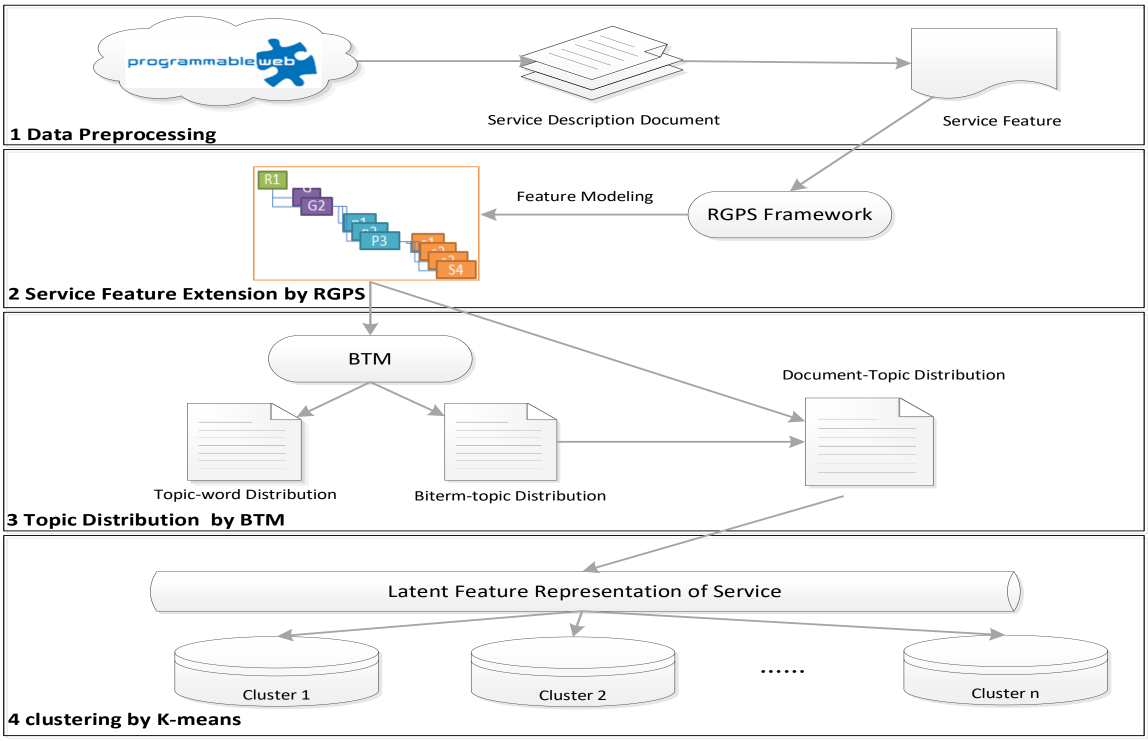
programmableweb
Service Description Document
Service Feature
1 Data Preprocessing
R1
G2
P3
S4
Feature Modeling
RGPS Framework
2 Service Feature Extension by RGPS
BTM
Document-Topic Distribution
Topic-word Distribution
Biterm-topic Distribution
3 Topic Distribution by BTM
Latent Feature Representation of Service
Cluster 1
Cluster 2
......
Cluster n
4 clustering by K-means

Figure 1. The framework of service clustering based on RGPS and BTM

Figure 1 illustrates the main process of RGPS and BTM based service topic clustering method, which is divided into four stages, namely data preprocessing, service feature extension, latent feature construction and service clustering. In the data preprocessing stage, the service data is crawled from the service registry PWeb, and the description of service is represented as the initial feature vector. In the stage of feature extension, RGPS is served to extend service features in four aspects, namely role, goal, process and service. In the latent feature construction stage, we obtain topic distribution by training BTM, and service is represented as a latent feature. In the stage of service clustering, the K-means algorithm is used to service cluster. The whole service clustering process is offline, so the performance of the algorithm can be guaranteed. In the experimental part, we mainly discuss the clustering performance of the algorithm and the influence of related parameters, similar to [24].

In this stage, the services are represented as the initial feature vector. The preprocess includes the following three steps:

(1) Feature vector construction: implement word segmentation of service description document based on NLTK [25].

(2) Word Stemming: extract the stem of feature words based on PorterStemmer algorithm in NLTK. For example, the stem of both “learned” and “learning” is “learn”.

(3) Stop word removal: eliminate stop word is to preserve the core words in the document, such as “the”, “a”, “and”, etc.

Applying the above steps, we will expand the service feature based on RGPS meta-model framework.

3 Feature Enhancement based on RGPS

The service feature extension must adhere to semantic content tag of RGPS, and it aims to enhance the semantic feature representation of the service. The feature extension analyzes the semantics of the service description document from four aspects. The specific process is shown in Figure 2, and M_r, M_g, M_p, M_s corresponds to role model, goal model, process model and service model respectively.

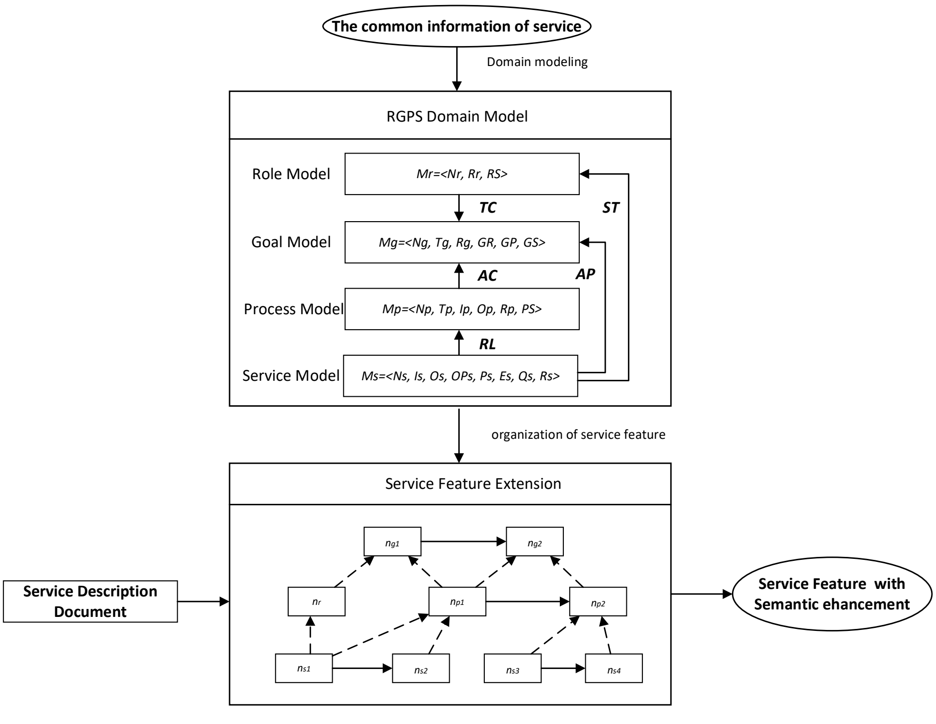
The common information of service
Domain modeling
RGPS Domain Model
Role Model
Mr=<Nr, Rr, RS>
TC
ST
Goal Model
Mg=<Ng, Tg, Rg, GR, GP, GS>
AC
AP
Process Model
Mp=<Np, Tp, Ip, Op, Rp, PS>
RL
Service Model
Ms=<Ns, Is, Os, OPs, Ps, Es, Qs, Rs>
organization of service feature
Service Feature Extension
Service Description Document
ng1
ng2
nr
np1
np2
ns1
ns2
ns3
ns4
Service Feature with Semantic ehancement

Figure 2. The extension framework of service feature based on RGPS

Definition 1: the role model describes a service’s consumer, provider or inquirer, etc, which can be defined as Equation (1):

$$M_r = \ < N_r, R_r, RS >. \tag{1}$$

N_r = { n_r| $n_r \in$ *Role* } is the set of role, where Role is the set of role’s name;

R_r = {< n_{ri}, r_r, n_{rj} >, n_{ri}, $n_{rj} \in N_r$, $r_r \in RS$} is the set of relationship between two roles,

$$RS \in \begin{Bmatrix} Included,\ Generalized,\ Extended, \\ Similar,\ Equivalent \end{Bmatrix};$$

RS = {<n_{ri}, ST, n_{sj}>, $n_{ri} \in N_r$, $n_{sj} \in M_s$, N_s} is the set of relationship between role and service, where *ST* means which services can satisfied the role’s requirement.

Definition 2: the goal model describes the goal of service, which can be defined as Equation (2):

$$M_g =< N_g, T_g, R_g, GR, GP, GS >. \tag{2}$$

$N_g = \{n_g | \ n_g \in Goal\}$ is the set of goal's name:

$T_g = \{t_g | \ t_g \in FG, NFG\}$ is the goal's category, and the classification includes Functional Goal (FG) and Non-functional Goal (NFG);

$GR = \{<n_{gi}, TC, n_{rj}>, n_{gi} \in N_g, n_{rj} \in M_r. N_r\}$ is the set of relationship between goal and role, where TC is the specific relationship;

$GP = \{<n_{gi}, AC, n_{pj}>, n_{gi} \in N_g, n_{pj} \in M_p. N_p\}$ is the set of relationship between goal and process, where AC is the specific relationship;

$GS = \{<n_{gi}, AP, n_{sj}>, n_{gi} \in N_g, n_{sj} \in M_s. N_s\}$ is the set of relationship between goal and service, where AP is the specific relationship.

Definition 3: the process model describes the service implementation, which can be defined as Equation (3):

$$M_p =< N_p, T_p, I_p, O_p, R_p, PS > . \quad \textbf{(3)}$$

$N_p = \{n_p | \ n_p \in Process\}$ is the set of process, where *Process* is the implementation of process;

$T_p = \{t_p | \ t_p \in AP, CP\}$ is the process's category, where AP is atomic process set, and CP is composition process set;

I_p is input, and O_p is output;

$R_p = \{<n_{pi}, r_p, n_{pj}>, n_{pi}, n_{pj} \in N_p, r_p \in RS\}$ is the structure of process;

$PS = \{<n_{pi}, RL, n_{sj}>, n_{pi} \in N_p, n_{sj} \in M_s. N_s\}$ is the set of relationship between process and service, where RL is the specific relationship.

Definition 4: the service model describes the specific service for implementing process, which can be defined as Equation (4):

$$M_s =< N_s, I_s, O_s, OP_s, P_s, E_s, Q_s, R_s > . \quad \textbf{(4)}$$

$N_s = \{n_s | \ n_s \in Service\}$ is the service's category, where *Service* is the service set;

I_s is input, and O_s is output;

$OP_s = \{op_s\}$ is the set of service operation;

P_s is the initial state, and E_s is the finial state;

Q_s is the information of QoS;

$R_s = \{<n_{si}, r_s, n_{sj}>, n_{si}, n_{sj} \in N_s, r_s \in RS\}$ is the relationship between two services, where RS is the specific relationship.

Definition 5: the feature model describes the extended service feature representation, which can be defined as:

$$CFM_s = \{CompCFM_s, GoalCFM_s, TripleCFM_s\} . \quad \textbf{(5)}$$

Where:

$CompCFM_s = \{e_i | \ e_i \in \{M_r. N_r, M_p. N_p, M_s. N_s\}\}$

$GoalCFM_s = \{e_g | \ e_g \in \{M_g. N_g\}\}$

$TripleCFM_s = \{<e_i, r_k, e_j> | \ e_i, e_j \in \{CompCFM_s, GoalCFM_s\}\}$

where $CompCFM_s$ is the element set of CFM_s, $GoalCFM_s$ is the goal set of CFM_s; and $TripleCFM_s$ is the whole relationship set.

Table 1 lists the detailed pseudo-code algorithm of the generation process of CFM_s, which can be used to guide service clustering.

Table 1. The service feature extension algorithm

Algorithm 1. Service feature extension algorithm based on RGPS

```
Input:  $M_r, M_g, M_p, M_s, WS = \{ws_1, ws_2, \dots, ws_n\}$
Output: $CFM_s$
1.          $CFM_s \leftarrow \emptyset, GoalCFM_s \leftarrow \emptyset$;
2.  For each $n_{g_i} \in \{M_g, N_g\}$
3.    $GoalCFM_s = GoalCFM_s \cup n_{g_i}$;
4.  End For
5.  For each $< n_{g_i}, TC, n_{r_j} > \in \{M_{g.} GR$
6.    If ($n_{r_j} \notin CFM_s$ ) Then
7.      $CompCFM_s \leftarrow CompCFM_s \cup n_{r_j}$;
8.      $T_s \leftarrow T_s \cup < GoalCFM_s . n_{g_i}, TC, CompCFM_s . n_{r_j} >$;
9.  End For
10. For each $<n_{g_i}, r_g, n_{r_j}> \in M_g, R_g$
11.     Find $GoalCFM_s . n_{g_i} == n_{g_i} \& GoalCFM_s . goal_j == goal$;
12.     $T_s \leftarrow T_s \cup < GoalCFM_s . n_{g_i}, r_g, CompCFM_s . n_{r_j} >$;
13.   End For
14.   Similar to step 10-13 and add role relations;
15.   For each $<n_{g_i}, r_g, n_{p_j}> \in \{M_{g.} G_p \&\& n_{p_j} \notin CFM_s$
16.     $CompCFM_s \leftarrow CompCFM_s \cup n_{p_j}$ ;
17.     $T_s \leftarrow T_s \cup < GoalCFM_s . n_{g_i}, AC, GoalCFM_s . n_{g_i} >$;
18.   For each $<n_{p_i}, RL, n_{s_j}> \in \{M_p. PS$
19.     If ($n_{p_i} == n_{p_j} \&\& n_{s_j} \notin CFM_s$ ) Then
20.     $CompCFM_s \leftarrow CompCFM_s \cup n_{s_j}$ ;
21.   End For
22.   For each service $ws_s \in WS$
23.     If ($sim\ (n_{s_j}, ws_s) > \alpha$ Then
24.     $cluster\ [k] \leftarrow cluster\ [k] \cup ws_s$;
25.     $T_s \leftarrow T_s \cup < CompCFM_s . n_{p_i}, RL, GoalCFM_s . n_{s_j} >$;
26.   End For
27.   Similar to step 10-13，add process and service relations;
28.   For each $<n_{g_i}, AP, n_{s_j}> \in M_g. GS$
29.     $T_s \leftarrow T_s \cup < GoalCFM_s . n_{g_i}, AP, GoalCFM_s . n_{s_j} >$;
30.   End For
31.   For each $<n_{r_i}, ST, n_{s_j}> \in M_r. RS$
32.     $T_s \leftarrow T_s \cup < GoalCFM_s . n_{r_i}, ST, GoalCFM_s . n_{s_j} >$;
33.   End For
34.   $SDP_{ws} \leftarrow CompCFM_s , T_s, GoalCFM_s$ );
35.   Return $CFM_s$
```

According to the sequence of Table 1, the specific role and goal are added to the model (steps 5~9), and the relation between the role and goal is added to the model (steps 10~13). The specific process and service are added to the model (steps 15~26), and the relationship among process, service and goal are integrated (steps 28~33). The output is the extended service feature CFM_s .

4 BTM Topic Model and K-means Algorithm

In this stage, we will elaborate on how to employ BTM to generate the service description document topic matrix, and then explain how to utilize the K-means to cluster and predict the cluster of service in the context of data sparsity.

4.1 BTM Topic Model

To tackle the sparsity problem during the service clustering, this paper proposes a bi-term based mixture model (BTM+K-means) which learns the topic over the service description document, and obtains the topic probability of documents using K-means algorithm.

BTM trains topic over short text by using the generation of bi-term in the corpus [14]. As we know, the service description document has little content. In all documents, we extract bi-terms to construct the corpus of BTM. For example, after the stop word removing and word stemming, the character “an AI autonomous vehicles” can be extracted as the following bi-terms: “AI autonomous”, “AI vehicle” and “autonomous vehicle”. After constructing the bi-terms, the number of words in each service document can be promoted to a reasonable amount, e.g. a document containing n words that are promoted to a document obtaining $\frac{n*(n-1)}{2}$ bi-terms, which can significantly alleviate data sparsity problem. The construction of bi-terms is formulated as follows: $B_d = \{(w_i, w_j)| w_i, w_j \in d, i \neq j\}$, and $B = \cup_{d \in D} B_d$. Here, B_d is the set of bi-terms which is extracted from the service document d, and each bi-term $b \in B_d$ contains two unordered words (w_i, w_j), D is the documents and d is the bi-terms, respectively.

BTM can construct the generation of bi-term with the latent topic structure, according to [26]. The extended service feature is used as training corpus B. Figure 3 illustrates the generative process of the corpus. Support each bi-term is drawn from a specific topic independently, the process can be shown as follows:

(1) For each topic $z = 1, 2, \ldots, n$, draw the topic-specific word distribution $\varphi_z \sim Dirichlet(\beta)$;

(2) Extract the global topic distribution for all short text collection $\theta \sim Dirichlet(\alpha)$;

(3) For each word pair b (w_i, w_j) in set B:

① Draw a topic assignment with $z \sim Multi(\theta)$;

② Draw w_i and w_j from topic z, namely, w_i, $w_j \sim Multi(\varphi_z)$.

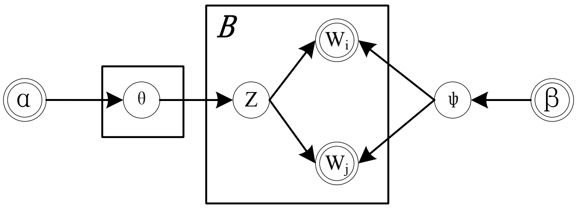

Figure 3. The generative graphical model of BTM

According to the corpus generation process, the joint probability of each bi-term can be calculated by Equation (5). Meanwhile, Equation (6) shows the probability of the whole corpus B.

$$P(b) = \sum_z P(z)P(w_i|z)P(w_j \mid z)\sum_z \theta_z \phi_{i|z} \phi_{j|z}. \tag{5}$$

$$P(B) = \prod_{(i,j)} \sum_z \theta_z \phi_{i|z} \phi_{j|z}. \tag{6}$$

From the graphical model of BTM, it can be seen that the subject of different bi-term in service description is independent, which is a significant difference between BTM and traditional topic model. We use the Bayesian assumption to perform approximate inference.

We can model the word pair pattern and get the topic probability of the document directly. Then, the document topic distribution should be calculated by Equation (7).

$$P(z \mid d) = \sum_b P(z|b)P(b|d). \tag{7}$$

The bi-term topic distribution $P(z|b)$ can be calculated by Bayesian formula as Equation (8).

$$P(z \mid b) = \frac{P(z)P(w_i \mid z)(P(w_j \mid z)}{\sum_z P(z)P(w_i \mid z)(P(w_j \mid z)}. \tag{8}$$

Where, $P(z) = \theta_z$ and $P(w_i|z) = \phi_{i|z}$.

In Equation (9), the empirical distribution of bi-term in the service description document can be used as the estimated value $P(b|d)$.

$$P(b \mid d) = \frac{n_d(b)}{\sum_b n_d(b)}. \tag{9}$$

Where, $n_d(b)$ is the frequency of bi-term b who appears in the document d.

In order to obtain the topic distribution of the service description document, we must estimate the value of parameter θ and φ. Commonly, the parameter estimation methods include expectation propagation, variation reasoning and Gibbs sampling, etc. In this paper, we choose Gibbs sampling to carry out posterior inference for the parameters study. Gibbs sampling is an applicable Markov Chain algorithm. With the core of Gibbs sampling, the parameters of θ and φ can be integrated out for conjugate prior distribution with the variables of α and β. The derivation process is described in the followings.

As the each word pair b (w_i, w_j) in set B, the posterior probability distribution is calculated to obtain the topic distribution z_b. Here, z_{-b} is the topic distribution except the word pair b in set B; n_z is the frequency that word pair b is assigned to topic z; $n_{w_i|z}$ is the frequency that word w_i is assigned to topic z; and M is the number of feature words as Equation (10).

$$P(z \mid z_{-b}, B, \alpha, \beta) \propto (n_z + \alpha)\frac{(n_{w_i|z} + \beta)(n_{w_j|z} + \beta)}{(\sum_w n_{w|z} + M\beta)^2}. \tag{10}$$

According to topic distribution of word pair, it is easy to calculate the value of parameter θ and φ as Equations (11) and (12).

$$\varphi_{w|z} = \frac{n_{w|z} + \beta}{\sum_{w} n_{w|z} + M\beta}. \tag{11}$$

$$\theta_z = \frac{n_z + \alpha}{|B| + K\alpha}. \tag{12}$$

Here, $\varphi_{w|z}$ is the probability of word w in the topic z; θ_z is the probability of topic z; and $|B|$ is the aggregated number of bi-term in set B.

4.2 K-means Algorithm

After mining the potential topics, K-means algorithm is adopted to cluster the topics, similar to [27]. K-means is a fast in memory algorithm, and it has the time complexity of $O(k^2x)$. It begins the clustering process by selecting k initial points as the temporary cluster center and assigning services to the cluster that they are closest to. Then the center of each cluster is taken as the new temporary center and services are re-assigned. These two steps are repeated until the changes in the center position fall below a threshold.

According to the above analysis, we calculate the service topic distribution $P(z|d)$by Equation (7). Many researches investigate service clustering based on service topic distribution. Chen et al [9] proposed to determine the clusters by Equation (13).

$$TC(S_i) = T_k \cap \forall j((j \neq k) \rightarrow P(S_i, T_j) < P(S_i, T_k)). \tag{13}$$

Where, $1 \leq i \leq K$, $1 \leq j \leq K$, $P(S_i, T_j)$ is the probability of the service S_i in the topic T_j, and the number of service topic is k, namely $T_1, T_2, \ldots, T_k$.

In this paper, each service is represented by Equation (14). Then we use K-means to cluster and obtain the service cluster information.

$$WS_d = [P(1|d), P(2|d), ..., P(k|d)]. \tag{14}$$

Table 2. The clustering algorithm

```
Algorithm 2.  K-means algorithm based on BTM
Input:  set of $CFM_s$, hyper-parameters $\alpha$, $\beta$, number of clusters
Output:  clusters of services
1.  Initialize $Z$, get the number of cluster $|Z|$;
2.  While Algorithm is not convergence Do
3.    For $iter = 1$ $iter = 1$ TO $Niter$
4.      For $b \in B$
5.        $P(z|z_{-b}, B, \alpha, \beta)$, sampling $z_b$;
6.        Replace $n_z$, $n_{w_i|z}$ and $n_{w_j|z}$;
7.      End For
8.    End For
9.  End While
10. Get parameter $\Theta$, $\Phi$;
11. According to (7)(8)(9),  build the topic-distribution, Then
12. Using K-means to cluster services;
13. Return clusters of services
```

The concrete steps of Semantic Sparse Service Clustering (S3C) are shown in Table 2. Here, Gibbs sampling is used to assist in obtaining the parameters (steps 1~9). Generated service topic distribution with Equations (7) to (9) is used to construct the latent feature of service (step 11). Afterwards, K-means algorithm is used to service cluster (step 12).

5 Experiments

We have conducted experiments to evaluate our proposed S3C method for short text clustering. To evaluate our method, this paper adopts the real-word Web service dataset PWeb. All experiments are implemented with JDK6.0 and Eclipse Helios Service Release.

5.1 Experimental Setup

Dataset. We use service registry PWeb as our dataset, and it has over 22,000 APIs. In our experiment, the service dataset consists of 2769 natural language descriptions of service API from 10 different application domains.

Baseline Methods. We designed a set experiment to compare our S3C model with three baseline methods. They are K-means, Agglomerative and LDA. As a result, 3356 distinct words have been extracted from dataset, and 1678 extended words have been added by RGPS framework.

5.2 Evaluation Metrics

We use *purity* and *entropy* to evaluate these four different methods. Let C_i be a cluster that contains n_i element, then the purity of each cluster and the average purity of all clusters are respectively defined as Equations (15) and (16).

$$P(c_i) = \frac{1}{n_i} \times \max_j (n_i^j). \tag{15}$$

$$purity = \sum_{i=1}^{k} \frac{n_i}{n} P(c_i). \tag{16}$$

Where, n_i^j is the number of correctly classified in clusters c_i that is in the classification of j.

Using symbols with the same meaning, the entropy of each cluster and the average of all clusters are respectively defined as Equations (17) and (18).

$$E(c_i) = -\frac{1}{\log(q)} \sum_{j=1}^{q} \frac{n_i^j}{n_i} \log(\frac{n_i^j}{n_i}). \tag{17}$$

$$entropy = \sum_{i=1}^{k} \frac{n_i}{n} E(c_i). \tag{18}$$

5.3 Results and Analysis

In addition to each topic cluster of service, the clustering results also include the probability between topic and feature words.

Table 3 illustrates the samples of the cluster in 5 topics, where the feature words are sorted according to the probability. For example, some words appear in more than one topic, such as “location” and “mobile”. The probability of “location” in topic cluster 1 is 0.078, while in topic cluster 4 is 0.178. The data is used to determine the service topic.

Table 3. The sample of word-topic distribution

Topic cluster1	Words	money	location	address	network	paypal	currency	payment	exchange	mobile	customer
	probility	0.389	0.078	0.054	0.071	0.274	0.196	0.234	0.322	0.036	0.089
Topic cluster2	Words	social	address	response	economy	software	location	platform	community	profile	map
	probility	0.186	0.093	0.075	0.034	0.078	0.178	0.263	0.369	0.154	0.112
Topic cluster3	Words	tool	ping	format	call	register	mobile	cart	order	map	\`network
	probility	0.456	0.239	0.211	0.354	0.147	0.093	0.012	0.064	0.124	0.357
Topic cluster4	Words	summarization	algorithm	research	\`precision	extraction	heuristic	overload	domain	publication	scheme
	probility	0.096	0.148	0.163	0.084	0.036	0.004	0.098	0.056	0.045	0.087
Topic cluster5	Words	support	account	auction	market	comment	method	ping	license	nerual	graph
	probility	0.075	0.036	0.126	0.098	0.298	0.178	0.035	0.188	0.093	0.013

For evaluate the clustering performance of S3C, it is compared with three classic service clustering methods.

K-means. It is a partition-based clustering method, and the similarity is measured by using the composite similarity [28].

Agglomerative. It is a typical hierarchical clustering method, and the similarity is measured by using the cosine similarity [29].

LDA. It is an unsupervised-topic clustering method, and the similarity is measured by KL divergence [30].

In the experimental results, Agglomerative method is worse than that of other methods. The reason is that if the data is sparsity, similarity may have too few ratings in common or may even give a negative correlation due to a small number of ratings.

Our method S3C improves data density by enhancing the service feature based on RGPS and BTM. Then our method clusters service by employing K-means algorithm. Moreover, our method systematically combines RGPS and BTM to predict the missing feature words and employs BTM and K-means to obtain the result of the service cluster, automatically. Thus, our method S3C achieves higher purity (lower entropy) than other methods. The results of the above methods are shown in Figure 4. The following conclusions are shown: 1) S3C is superior to other algorithms in two evaluation indexes, and the performance is significantly improved. 2) The performance of K-means, agglomerative and LDA is not good when it is met in the situation of data sparsity.

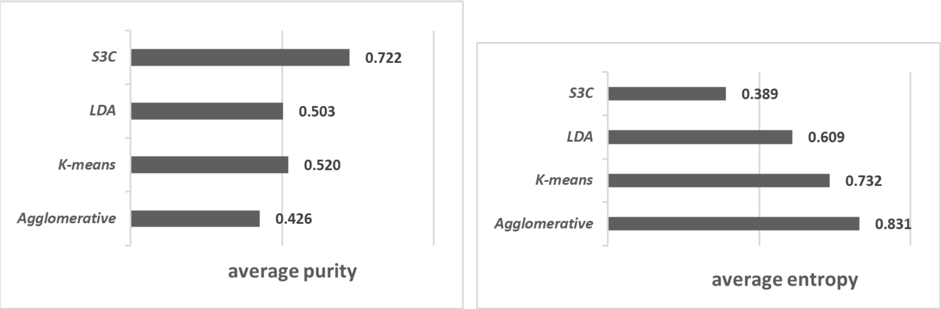

Figure 4. The performance comparison of four methods

5.4 Parameter Setting

We observe the impact of parameters, including the number of service feature word (i.e., n), and the number of service clustering (i.e., k). Both purity and entropy can reach the best value when $n = 160$, $k = 60$, respectively.

In order to verify feature enhancement effectiveness to alleviate the data sparsity problem and justify the usage of RGPS framework, we conduct experiments on the feature word number n with 80, 120, 160, 200 and 240 respectively. The feature word number n directly influences service clustering. We compare the average purity and entropy of our method with some other famous methods under different feature words n. We change the number of feature words from 80 to 240 with a step value of 40. When n is 160, the experimental results achieve higher purity and lower entropy. Therefore, we expand the number of feature words to about 160. Thus, a proper parameter value for *n* is very important.

BTM is an unsupervised topic model, and the number of topic must be set before modeling. The number of topics is an important factor affecting the performance of model. Based on the Bayesian Selection approach, the topic number is determined by Gibbs sampling calculation. Gibbs sampling is running under different values of k to detect the change of $\log(P(w|K))$. Since different value of k can affect the extraction effects, the experimental value of K is set with 30, 40, 50, 60, 70, 80, 90 respectively and the number of iterations is 1000, $\alpha = 50/\mathrm{k}$, $\beta = 0.01$. The performance of S3C method by varying the value of β. We find that the best result is achieved when β is about 0.01. When increasing the value of β, the performance of S3C increases first, and declines finally. The number of service cluster drops quickly when increasing β. The K-means result is influenced by the initial center, and we obtain the results by 20 times of experiments. When the value of k is 60, the posterior probability can obtain the best performance, and the topic model achieves the best fitting degree by the given data.

BTM+K-means are compared to traditional K-Means, and our method is slightly better than another, which is mainly because the keyword holds a high weight in the service text, and the keyword can be more representative of the topics. The results show that the service feature after semantic enhancement can better describe the information of the service.

6. Conclusion

We propose the Semantic Sparse Service Clustering (S3C), a novel service clustering method, which is effective in semantic enhancement and intuitive to interpret. This paper proposes a topic model for service clustering in the context of data sparsity. The main process of S3C is divided into four stages: data crawling and preprocessing, RGPS-based feature extension, latent feature construction and service clustering. Through the experiments, it is tested that our method can effectively alleviate the problem of semantic sparseness by expanding the service description document from multiple dimensions based on RGPS meta-model framework and obtaining a better topic distribution effect by constructing the latent feature of service description in BTM. The comparative experiment performed on PWeb dataset demonstrates the effectiveness of our method and shows that the proposed method significantly improves accuracy of service clustering.

We will apply our service clustering method to long documents to observe whether the semantic enhancement leads to performance improvement. We plan to use clustering for computing service similarity as so to get the better service recommendations. We extract the words in the keywords of the web service description and vectorize these feature words based on Word2Vec. The web service description vector is obtained and performs clustering on all web service description vectors to obtain k web service clusters for service recommendation.

7 Acknowledgements

This work is supported by the Natural Science Foundation of Hubei Province under Grant No. 2020CFB807; the Key Project of Hubei Education Department under Grant No. D20201402 & D20191406; the Teaching Research Project of Hubei Education Department under Grant No. 2021295; the Science Start-up Foundation for High-level Talents of HBUT under Grant No. 430100391 & 337396.

References

[1] C. Xu, D. Li, W. E. Wong, M. Zhao, Service Caching Strategy based on Edge Computing and Reinforcement Learning, *International Journal of Performability Engineering*, Vol. 18, No. 5, pp. 350-358, May, 2022.

[2] B. Xia, Y. Fan, W. Tan, K. Huang, J. Zhang, C. Wu, Category-aware API Clustering and Distributed Recommendation for Automatic Mashup Creation, *IEEE Transactions on Services Computing*, Vol. 8, No. 5, 674-687, September-October, 2015.

[3] K. Elgazzar, A. Hassan, P. Martin, Clustering WSDL Documents to Bootstrap the Discovery of Web Services, *in Proc. of IEEE International Conference on Web Services*, Miami, Florida, USA, 2010, pp. 147-154.

[4] H. Li, D. Li, W. E. Wong, D. Zeng, M. Zhao, Kubernetes Virtual Warehouse Placement based on Reinforcement Learning, *International Journal of Performability Engineering*, Vol. 17, No. 7, pp. 579-588, July, 2021.

[5] J.-L. Chen, N.-O. Hembara, M. Hvozdyuk, Nonstationary Temperature Problem for a Cylindrical Shell with Multilayer Thin Coatings, *Materials Science*, Vol. 54, No. 3, pp. 339-349, November, 2018.

[6] J. Tang, X. Wang, H. Gao, X. Hu, H. Liu, Enriching Short Text Representation in Microblog for Clustering, *Frontiers of Computer Science*, Vol. 6, No. 1, pp. 88-101, February, 2012.

[7] I. H. Witten, E. Frank, M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, *Morgan Kaufmann*, 2016.

[8] J. Wu, L. Chen, Z. Zhen, M. R. Lyu, Z. Wu, Clustering Web Services to Facilitate Service Discovery, *Knowledge and information systems*, Vol. 38, No. 1, pp. 207-229, January, 2014.

[9] L. Chen, Y. Wang, Q. Yu, Z. Zheng, J. Wu, WT-LDA: user tagging augmented LDA for web service clustering, *Proc. of the 2013 International Conference on Service-Oriented Computing*, Berlin, Germany, pp. 162-176, 2013.

[10] F. Xie, J. Wang, R. Xiong, N. Zhang, Y. Ma, K. He, An Integrated Service Recommendation Approach for Service-based System Development, *Expert Systems with Applications*, Vol. 123, No. 6, pp. 178-194, June, 2019.

[11] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent Dirichlet Allocation, *The Journal of Machine Learning Research*, Vol. 3, pp. 993-1022, March, 2003.

[12] Z. Gao, Y. Song, S. Liu, H. Wang, H. Wei, Y. Chen, W.

Cui, Tracking and Connecting Topics via Incremental Hierarchical Dirichlet Processes, *Proc. of 11th International Conference on Data Mining,* Vancouver, BC, Canada, 2011, pp. 1056-1061.

[13] B. Q. Cao, X. Q. Liu, B. Li, MD M. Rahman, B. Li, J. X. Liu, M. D. Tang, Integrated Content and Network-Based Service Clustering and Web APIs Recommendation for Mashup Development, *IEEE Transactions on Services Computing*, Vol. 13, No. 1, pp. 99-113, January-February, 2020.

[14] X. Yan, J. Guo, Y. Lan, X. Cheng, A Biterm Topic Model for Short Texts, *Proceedings of the 22nd international conference on World Wide Web*, Rio de Janeiro Brazil, 2013, pp. 1445-1446.

[15] X. H. Phan, L. M. Nguyen, S. Horiguchi, Learning to Classify Short and Sparse Text & Web with Hidden Topics from Large-scale Data Collections, *Proc. of the 17th International Conference on World Wide Web*, Beijing, China, 2008, pp. 91-100.

[16] M. Cataldi, L. Di Caro, C. Schifancella, Emerging Topic Detection on Twitter based on Temporal and Social Terms Evaluation, *Proc. of the 10th International Workshop on Multimedia Data Mining*, Washington, D.C, USA, 2010, Article No. 4.

[17] J. X. Liu, K. Q. He, J. Wang, D.-H. Yu, Z.-W. Feng, Da Ning, X.-W. Zhang, An Approach of RGPS-Guided On-demand Service Organization and Recommendation *Chinese Journal of Computers*, Vol. 36, No. 2, pp. 238-251, February, 2013.

[18] R. Lu, L. Xiang, M. R. Liu, Q. Yang, Discovering News Topics from Microblogs based on Hidden Topics Analysis and Text Clustering, *Pattern Recognition & Artificial Intelligence*, Vol. 25, No. 3, pp. 382-387, June, 2012.

[19] J. Xu, B. Xu, P. Wang, S. Zheng, G. Tian, J. Zhao, B. Xu, Self-taught Convolutional Neural Networks for Short Text Clustering, *Neural Networks*, Vol. 88, pp. 22-31, April, 2017.

[20] A. Hadifar, L. Sterckx, T. Demeester, C. Develder, A Self-training Approach for Short Text Clustering, *Proc. of the 4th Workshop on Representation Learning for NLP*, Florence, Italy, 2019, pp. 194-199.

[21] B. Zhang, B. Peng, J. Qiu, High Performance LDA through Collective Model Communication Optimization, *Procedia Computer Science*, Vol. 80, pp. 86-97, 2016.

[22] H. Xie, H. Jiang, Improved LDA model for microblog topic mining, *Journal of East China Normal University (Natural Sciences)*, No. 6, pp. 93-101, 2013.

[23] J. L. Chen, J. Su, O. Kochan, M. Levkiv, Metrological Software Test for Simulating the Method of Determining the Thermocouple Error in Situ During Operation, *Measurement Science Review*, Vol. 18, No. 2, pp. 52-58, April, 2018.

[24] M. H. Wu, H. H. Yue, J. Wang, Y. X. Huang, M. Liu, Y. H. Jiang, C. Ke, C. Zeng, Object Detection based on RGC Mask R-CNN, *IET Image Processing*, Vol. 14, No. 8, pp. 1502-1508, June, 2020.

[25] H. Zhang, G. Zhong, Improving Short Text Classification by Learning Vector Representations of Both Words and Hidden Topics, *Knowledge-Based Systems*, Vol. 102, pp. 76-86, June, 2016.

[26] J. L. Chen, V. Yatskiv, A. Sachenko, J. Su, Wireless Sensor Networks based on Modular Arithmetic, *Radioelectronics & Communications Systems*, Vol. 60, No. 5, pp. 215-224, May, 2017.

[27] M. H. Wu, R. Chen, Y. Tong, Shadow Elimination Algorithm using Color and Texture Features, *Computational Intelligence and Neuroscience*, Vol. 2020, pp. 1-10, January, 2020.

[28] Z. Hong, Q. Shao, X. Liao, R. Beyah, A Secure Routing Protocol with Regional Partitioned Clustering and Beta Trust Management in Smart Home, *Wireless Networks*, Vol. 25, No. 7, pp. 3805-3823, October, 2019.

[29] J. W. Wu, J. C. R. Tseng, W. N. Tsai, A Hybrid Linear Text Segmentation Algorithm using Hierarchical Agglomerative Clustering and Discrete Particle Swarm Optimization, *Integrated Computer-Aided Engineering*, Vol. 21, No. 1, pp. 35-46, 2014.

[30] T. Scheffler, R. Schirru, P. Lehmann, Matching Points of Interest from Different Social Networking Sites, *German Conference on Advances in Artificial Intelligence*, Saarbrücken, Germany, 2012, pp. 245-248.

Biographies

Fang Xie received the PhD degree in computer science and technology from Wuhan University. She is a lecture in Computer Science School, Hubei University of Technology. Her research focuses on big data processing, and service computing.

Jing-Liang Chen received the PhD degree in computer science and technology from Wuhan University. He is an associate professor in Computer Science School, Hubei University of Technology. His research focuses on big data processing and machine learning.

Yi Zhu received the PhD degree in computer software from School of Computer Science, Huazhong University of Science and Technology. She is a lecturer in computer network technology, Hubei Communications Technical College. Her research focuses on database security and cloud computing.

Hong-Yan Zheng received the PhD degree in army command from Rocket Force Command College. She is an associate professor in Computer Science Technology, National University of Defense Technology. Her research focuses on target recognition and UAV route planning.