Relationship Network Augmented Web Services Clustering

Yingcheng Cao[1,2], Jianxun Liu[1,2,*], Min Shi[1,2], Buqing Cao[1,2], Xiangping Zhang[1,2], Yan Wang[3]
[1]School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China
[2]Key Laboratory of Knowledge Processing & Networked Manufacturing, Hunan University of Science and Technology
[3]Department of Computing, Macquarie University, Sydney, NSW 2109, Australia
caoyingcheng12138@gmail.com, ljx529@gmail.com, yan.wang@mq.edu.au

Abstract — **Clustering Web services can promote the quality of services discovery and management within a service repository. Traditional clustering methods primarily focus on using the semantic distance between service features, *i.e.*, latent topics learned from WSDL documents, to measure the service content similarity between Web services. Few works exploited the structural information generated during the usage of Web services, *i.e.*, the service compositing and tagging behaviors. Nowadays, Web services frequently interact (e.g., composition relation and tag sharing relation) with each other to form a complex service relationship network. The rich network relations inherently reflect either positive or negative categorical relevance between services, which can be strong supplement of service semantics in characterizing the functional affinities between services. In this paper, we propose to utilize the services relationship network for augmented services clustering algorithm design. We first learn semantic information from service descriptions based on the widely used Doc2Vec model. Then, we propose a revised K-means algorithm for service clustering that benefits simultaneously from service semantics and network relations, where the service relations are previously preserved in a set of low-dimensional vectors achieved based on a recently proposed network embedding technique. Experiments on a real-world dataset demonstrated that the proposed clustering approach yields an improvement of 6.89% than the state-of-the-art.**

Keywords- Web services; services clustering; services network; network embedding; Semantic mining

I. INTRODUCTION

With the development of Web 2.0 technologies, the past decade has witnessed a rapid growth of Web services and their compositions (e.g. Mashups) on the Internet [1]. The increasing amount of Web services have greatly added the burden of people using and managing the service repository efficiently [2]. In this past, Web services clustering has been demonstrated to be an effective technique to mitigate this challenge [3], [4]. As an important tool for exploratory data analysis, Web services clustering can help us capture and understand the hierarchical functional structures of services within a closed repository such as the ProgrammableWeb. It then greatly facilitates a wide range of downstream tasks, such as services discovery or selection [3], [4], [5], [6], services replication [7], services composition/Mashup [8], services recommendation [9], [10], [11], [12], and etc.

A lot of related works on service clustering have been done so far. Most of them focus on mining the service semantics [13], [14] based on some probabilistic topic models such as LDA [15] and HDP [16]. They typically first learn latent topic vectors to represent service description documents. Affinities between services are then measured by the cosine similarity between these vectors. However, a major issue with these methods is that they have been based on only the plain and static service content information, which might lead to suboptimal performance since service descriptions are usually short, sparse and littered with noisy features [19]. Therefore, some investigations [17], [18], [19], [20] seek to incorporate auxiliary information, such as tags and prior knowledge into the clustering process. However, these methods only exploit either service contents or service connectivity structures as plain genes to reduce the effect of data sparsity problem. For example, tags were introduced to improve the LDA training process [17], [18]. Nevertheless, tags are created by different service developers that may bring the vocabulary gap problem (e.g., two different words express similar meaning), causing semantic inconsistency among functionally equivalent services. As we know, services are not only created to run independently, but also cooperate with others to accomplish complex tasks, *i.e.*, composing services with diverse functionalities to form some value-added applications (a.k.a Mashups). The frequent interconnection (composition or tag sharing relations) between services naturally form a complex relationship network that actually reveal the semantic relevance among services. The formation of the services relationship network comes from users' usage of services (e.g., compositing and tagging behaviors). In fact, these behaviors analogical to the folksonomy can be seen as manual processes of classifying Web services. For example, tagging a service means classifying it to a corresponding functional category according to the user's interpretation, and services with similar tags should likely belong to the identical clusters. Similarly, composition behaviors mean that users seek to aggregate multiple functional complementary services of different categories together. We argue that these positive and negative categorical relevance derived from the network relations can be used to enhance the semantic-based Web service clustering process.

Fig. 1 illustrates the aforementioned two common behaviors of service users, visualized by the composition relation (red arrow) between Mashups and services and tagging relation (blue arrow) between services and tags. The composition relation represents that Web services are once being gathered together, *i.e.*, to be invoked by a specific Mashup. For example, services "YouTube" and "Google ma-

978-1-7281-2717-0/19/$31.00 ©2019 IEEE
DOI 10.1109/ICWS.2019.00050

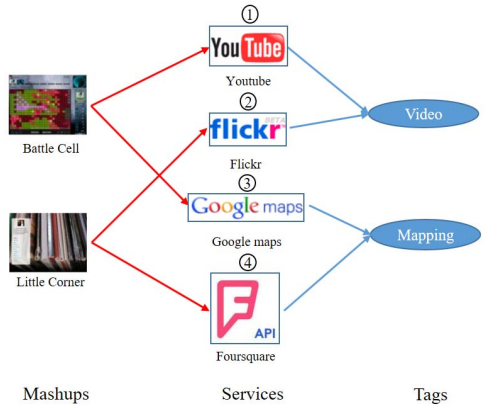


Figure 1. A snapshot of a service relationship network.

ps" as well as services "Flickr" and "foursquare" in Fig. 1 possess composition relationship. Web services used here stand for API services in the rest of this paper. Tagging relation denotes that a tag is annotated to various Web services, *i.e.*, services "Flickr" and "YouTube" possess tagging relation since they are both labeled by the same tag "Video". From these network relations, we can derive some helpful information to the Web service clustering. For example, as both "YouTube" and "Flickr" are tagged with "Video", we can assume that they both provide functionality related to "Video", which means they have similar functions. They are then most likely to be clustered into the same functional domain. In terms of relationship, when a user develops a mashup that meets a range of functional requirements, he/she needs to call APIs from diverse functional domains. Therefore, it is very possible that those services being composed together belong to different functional domains. However, it is non-trivial to simultaneously incorporate the service semantics and relation structures for unified service feature representation learning, which is critical for optimal service clustering performance. In this paper, from the content perspective, we first learn the semantic vectors from service descriptions based on the Doc2vec model [21]. Meanwhile, from the network structure perspective, we propose to preserve the service network relations by a set of low-dimensional vectors achieved based on a recently proposed network embedding technique [22]. Finally, we propose a revised K-means algorithm for augmented Web service clustering that benefit simultaneously from above two-aspect feature vector learning. Specifically, our contributions are summarized as follows:

- We analyzed and proposed to combine service semantics and service network structures for services clustering targeted at large-scale service repository management. To enable the integration, we introduce two recently proposed models to learn clustering features from the service descriptions and network relations, respectively.
- We proposed a K-means algorithm for service clustering that can seamlessly incorporate the semantic and structural features, where the network relations can enhance and refine the affinities measurement between services calculated based on the service semantics.
- We designed extensive experiments to evaluate the proposed approach on a real-world dataset. The experimental results demonstrate that network structure information can be strong supplement to service content in performance gain, i.e., the denser of the service network connectivity, the more accurate of the clustering results.

II. PRELIMINARY KNOWLEDGE

To better understand the idea of the proposed approach, this section presents some preliminary knowledge.

A. Notations and Problem Formulation

Formally, Web services are denoted as $V = \{v_1, v_2, v_3, \ldots v_n\}$, the description document of a Web service a is represented as $D^{(a)} = \{w_1, w_2, \ldots, w_n\}$, where w_i is the ith word of the document, and n is the number of words that the document contains. Tags annotated with Web service a are defined as $T^{(a)} = \{t_1, t_2, \ldots, t_m\}$, where t_i is the ith tag of the Web service, and m is the number of tags.

The task of the proposed approach is to cluster all Web services into different service domains according to their functionalities.

B. Services Relationship Network

As discussed in the previous section, there are two types of service relations that might be helpful but exert different impacts on the clustering closeness measurement, *i.e.*, services with shared tags tend to be centralized while services being composed together should be away from each other. Motivated by these assumptions, we can construct a signed service relationship network shown in Fig. 2, with positive and negative signs on edges to manifest the categorical relevance between two corresponding Web services. To better describe the algorithms later, we represent such a service relationship network as $G=(V, E)$, where $V=\{v_i\}_{i=1,\ldots N}$ is a set of Web service nodes, $e_{i,j}= (v_i, v_j) \in E$ is an edge reflecting the edge relation between two services, where $e_{i,j} = 1$ means positive link between vi and v_j (e.g., annotation relationship), and $e_{i,j}=-1$ denotes negative link (e.g., composition relationship). In this paper, if two services share at least 1 tag, then a positive link will be built. In comparison, if two services are being composed together, then a negative sign will be given. Based on the dataset used in this paper, the produced signed service network finally contains 6718 vertices, 4031 negative links and 51525 positive links, respectively.

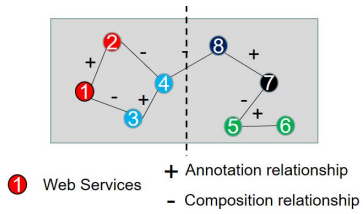


Figure 2. An example network schema of services relationship network (left part is from Fig. 1).

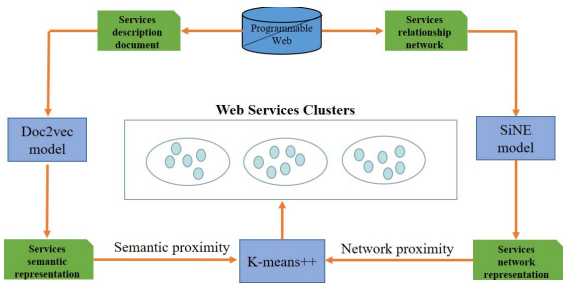


Figure 3. The Framework of our Web services clustering approach.

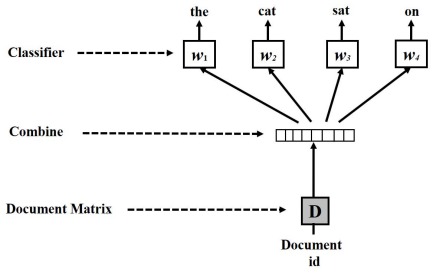


Figure 4. Framework of PV-DBOW Model.

III. THE METHODOLOGY

The overall framework of our approach is shown in Fig. 3. First of all, we learn the semantic representations of service description documents based on the Doc2Vec model [21]. Meanwhile, we learn the network relations feature representations based on the SiNE model [23], where the signed service relation network is previously constructed according to the compositing and tag sharing relations among services. Finally, all Web services are clustered into different functional domains based on a revised K-means algorithm which takes as inputs the semantic representations and network representations.

A. Service-Semantics Representations Module

From the content perspective, many existing methods represent each Web Service with a vector as the input of some clustering algorithms. Early studies employ bag of word approaches [4] (e.g., TF-IDF) or topic models [15] (e.g. LDA) to represent each document with a vector. However, these models do not consider the context information of a document (i.e., order of words), resulting in suboptimal representation due to the short service description texts. To model semantic similarities between services, in this paper, we employ the state-of-the-art approach called Doc2Vec to learn the latent semantic vectors of Web services description documents. Fig. 4 shows its learning mechanism, which assumes that each document is responsible for inferring all its included words. Such a paradigm would make documents having many shared words to also have close semantic representations. Assume the description word sequence of service v_i is represented as $\{w_1, w_2, \ldots, w_N\}$, then the representation learning is to maximize the following objective function:

$$\mathcal{L} = \sum_{i=1}^{N} \log \mathbb{P}(w_{-b}: w_b | v_i) \quad (1)$$

where $w_{-b}: w_b$ is a sequence of words inside a contextual window of length b.

B. Service-Relations Representations Module

Network representation learning has been recently proposed as a new learning paradigm to embed network vertices into a low-dimensional vector space while being preserved well the network topology structure, the vertex content, and other side information. This facilitates the original network to be easily handled in the new vector space for further analysis [22]. To represent networks in different scenarios, many network embedding methods are proposed, e.g. DeepWalk [24] and node2vec [25]. In this paper, we focus on a recently proposed network embedding approach called SiNE [23]. It is a deep neural network-based model for learning signed social networks, which are demonstrated to be effective on a variety of social media sites, such as Epinions with trust and distrust links, and Slashdot with friend and foe links. Based on the structural balance theory, nodes should be closer to their friends (linked with positive edges) than their foes (linked with negative edges). SiNE preserves this property by maximizing the margin between the embedding proximity of friends and the embedding proximity of foes.

Let P be a set of triplets (v_i; v_j; v_k) as shown in Fig. 5(a) from the network, where v_i and v_j have a positive link while v_i and v_k have a negative link. Formally, P is defined as:

$$P = \{(v_i, v_j, v_k) | e_{ij} = 1, e_{ik} = -1, v_i, v_j, v_k \in V\} \quad (2)$$

The extended structural balance theory suggests that with a certain similarity measurement, for a triplet $(v_i, v_j, v_k) \in P$,

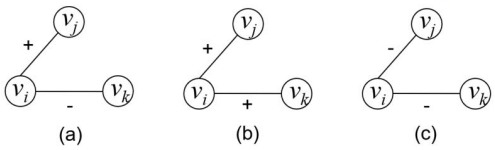


Figure 5. Three Types of Triplets of Nodes in the Network.

v_i is likely to be more similar to the user with a positive link, i.e. v_j, than a user with a negative link, i.e. v_k, which can be mathematically modeled as:

$$f(x_i, x_j) \geq f(x_i, x_k) + \delta \quad (3)$$

where x_i, x_j and x_k are the d-dimensional vector representations of v_i, v_j and v_k respectively, which we need to learn by the embedding framework. In $f(x_i, x_j)$, f is a function that measures the similarity between x_i and x_j. The parameter δ is a threshold that is used to regulate the difference between these two similarities. A large δ will force v_i, v_j to be closer and v_i, v_k to be dissimilar. Due to the space limitation, please refer to the literature where the SiNE [23] is proposed.

We use SiNE model to learn a low-dimensional vector $R_i \in \mathbb{R}^d$ (d is a smaller number) for each service v_i in the network. Therefore, services close to each other in the network topology or connected with positive links (*e.g.*,

annotation relationship) are close in the representation space. On the contrary, services with composition relations are far away in the representation space as shown in Fig. 6. The visualization experiment in the next section verifies our point of view.

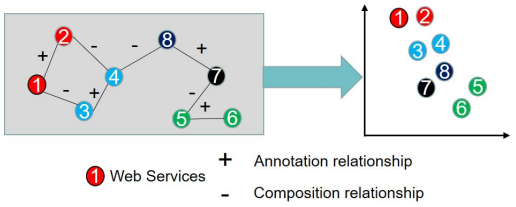


Figure 6. 2D visualization on services relationship network by SiNE.

C. Services-Clustering Module

The services clustering module is the core part of our model, in which services relationship information and services semantic information are integrated. We first elicit the latent semantic information of Web service description documents based on the Doc2Vec model. S_i= (s_{i1}, s_{i2}, s_{i3}, ..., s_{id}) stands for the semantic vector of service v_i, d is the dimension of the vector. Then, given a services relationship network, we use the network embedding model SiNE to obtain the vector format of each service. R_i= (r_{i1}, r_{i2}, r_{i3}, ..., r_{id}) stands for the relationship vector of service v_i, which can describe the relationship among the services and reflects the overall structure of the network. The dimension of R_i is also set to be d. Finally, based on above obtained latent semantic vectors of description documents and services relationship network, we perform the Web services clustering process based on a revised K-means algorithm. We aim to partition these Web services into h clusters C= {C_1, C_2, C_3, ..., C_h} so as to minimize the objective function:

$$\Gamma = \sum_{j}^{h} \sum_{i=1, v_i \in C_j}^{n} d(v_i, u_j)^2 \quad (4)$$

where u_j is the mean of the points in C_j. We adopt the following modified Euclidean distance (Eq. 7), which incorporates information from two parties: services semantic information and services relationship information to jointly calculate the distance between v_i and the cluster center u_j during the clustering process. The clustering process is performed at two levels: (1) at the services content level, our model captures inter-service relation by maximizing the co-occurrence of word sequence given a service description document (Eq. 5); (2) at the services relationship network level, our model exploits cross-service relationship by maximizing the probability of observing similar services given a service in the constructed network (Eq. 6);

$$d(S_i, S_j) = \frac{\sum_{k=1}^{t} s_{ik} s_{jk}}{\sqrt{\sum_{k=1}^{n} s_{ik}^2} \sqrt{\sum_{k=1}^{n} s_{jk}^2}} \quad (5)$$

where S_i is the semantic vector of current service v_i, and S_j is the mean semantic vector of the services in cluster C_j. d (S_i, S_j) represents the document-level similarity between the current service and other services in cluster C_j in the services semantic vector space.

$$d(R_i, R_j) = \frac{\sum_{k=1}^{t} r_{ik} r_{jk}}{\sqrt{\sum_{k=1}^{t} r_{ik}^2} \sqrt{\sum_{k=1}^{t} r_{jk}^2}} \quad (6)$$

where R_i is the relationship vector of current service v_i, and R_j is the mean relationship vector of services in cluster C_j. Here, d (R_i, R_j) is the cosine similarity between the current service and other services in cluster C_j in the service relationship vector space. d (R_i, R_j) represents the extent to which the current service is related to other services in the cluster.

$$d(v_i, u_j) = \alpha \times d(S_i, S_j) + (1 - \alpha) \times d(R_i, R_j) \quad (7)$$

where α is a trade-off parameter that balances services semantic information and services relationship information. Hyper-parameter investigation and impact of α will be described and discussed in the next section.

Therefore, the services relationship information and services semantic information is able to influence Web services clustering process simultaneously. The Doc2Vec-RK algorithm for clustering Web services is presented with details in Table I.

TABLE I. THE DOC2VEC-RK ALGORITHM

Algorithm 1: The Doc2Vec-RK algorithm

Input: *a collection of cluster points P*
Input: *the services relationship network previously constructed*
Output: *Disjoint h partitioning of* C= $U_{j=1}^{h} \boldsymbol{C_j}$ *such that the K-means objective function is optimized*
Process:
1 construct a set U of h initial centers, choose the first point $\boldsymbol{u_1}$ uniformly at random from the cluster points P and add it to U
2 **repeat**
3 choose the next added center $\boldsymbol{u_j}$,selecting $\boldsymbol{u_j} = \boldsymbol{v_i} \in P$ with probability $\frac{D(v_i)^2}{\sum_x^P D(v_x)^2}$,where D($\boldsymbol{v_i}$) denotes the shortest distance form $\boldsymbol{v_i}$ to its closest center in U, calculated based on Equation (7)
4 **until** (the set U contains h initial centers)
5 **repeat**
6 assign each data point $\boldsymbol{v_i} \in P$ to the cluster $\boldsymbol{u_j} \in U$ where $\boldsymbol{v_i}$ has the shortest distance with $\boldsymbol{u_j}$ based on Equation (7).
7 update the cluster centers:$\boldsymbol{u_j} \leftarrow \frac{1}{|C_j|} \sum_{v_i \in C_j} \boldsymbol{v_i}$
8 **until** (convergence)

After above clustering process, all Web services are clustered into different functional groups according to their semantic information and relationship information. Besides the cosine-based similarity in the Eq. 5, it is feasible to use other similarity measures, such as Jaccard similarity, to measure the closeness between services. In the next section, a comprehensive evaluation is conducted to validate the performance of the proposed approach.

IV. EVALUATION

A. Dataset Description

To evaluate the proposed approach, we crawled a Web service dataset from ProgrammableWeb.com and finally obtained 6206 real Mashup services, 12919 API services as well as their related information. This crawled dataset is available at http://kpnm.hnust.cn/xstdset.html. There are totally 384 categories for 12919 Web services and the average size of each category is 33.73. The number of Web services in each category is severely uneven. For example, the category Tools contains 790 Web services while the category law contains 1 service only. As categories with less Web services would result in poor clustering performance, we only choose here the top 20 categories, which involve 6718 Web services, as our experiment dataset. Table II shows the detailed distribution data of the top 20 Web services categories.

TABLE II. THE DISTRIBUTION OF WEB SERVICES IN TOP 20 CATEGORIES

Category	Number	Category	Number
Tools	790	Telephony	285
Financial	586	Reference	278
Enterprise	487	Advertising	248
eCommerce	435	Email	240
Social	403	Travel	237
Messaging	388	Search	234
Payments	374	Video	216
Government	306	Security	216
Mapping	295	Education	208
Science	287	Transportation	205

B. Evaluation Metrics

In our experiments, we evaluate the clustering performance by three metrics, *i.e.*, Precision, Recall, and Purity. Suppose the standard classification of Web services in top M categories as $RSC = \{RC_1, RC_2, ..., RC_M\}$, which is available in the crawled dataset. We represent the experimental Web services clustering results as $ESC = \{EC_1, EC_2, ..., EC_V\}$. The Precision and Recall metrics are defined as follows:

$$Recall(EC_i) = \frac{|EC_i \cap RC_i|}{|RC_i|} \quad (8)$$

$$Precision(EC_i) = \frac{|EC_i \cap RC_i|}{|EC_i|} \quad (9)$$

Where $|EC_i|$ the number of Web services in cluster EC_i, $|RC_i|$ is the number of Web services in RC_i and $|EC_i \cap RC_i|$ is the number of Web services in successfully placed into cluster RC_i.

The Purity of cluster EC_i and the mean Purity of *ESC* are defined as follows:

$$Purity(EC_i) = \frac{max_j|EC_i \cap RC_j|}{|EC_i|}, 1 \le j \le M \quad (10)$$

$$Purity(ESC) = \sum_{i=1}^{TK} \frac{|EC_i|}{N} \times Purity(EC_i) \quad (11)$$

where N is the total number of Web services in *RSC*, and *TK* represents the top k $(1 \le k \le V)$ clusters in the experiment.

For each cluster EC_i,we calculate the Recall, Precision and Purity, respectively. The average results are finally reported.

C. Performance Comparison

In this section, we compare our approach with the following approaches to verify the effectiveness of the proposed approach.

1) **TFIDF-K [4]**: This method adopts the K-means algorithm to cluster Web services. The similarity calculation between services is based on the term frequency and inverse document frequency (TF-IDF)
2) **LDA [15]**: LDA model is used to cluster Web services. Each service belongs to a unique topic with maximum topic probability value. Then Web services have the same assigned topics are clustered together.
3) **Doc2Vec-K [21]**: This method adopts the K-means algorithm to cluster Web services. The similarity calculation between services is based on their probability topic vectors pre-trained by the Doc2Vec model.
4) **LDA-K [13]**: This method adopts the K-means algorithm to cluster Web services. The similarity calculation between services is based on their probability topic vectors pre-trained by the LDA model.
5) **LDA-PK [20]**: It proposes a Web service clustering method by leveraging prior knowledge to enhance the clustering process in a semi-supervised way. The similarity calculation between services is based on some prior knowledge and their latent topic information obtained by LDA model.
6) **Doc2Vec-RK**: The method is proposed in this paper. It utilizes both services relationship information and services semantic information simultaneously to promote Web services clustering process. The similarity calculation between services is based on their semantics obtained by Doc2Vec model and relations of services.

Since we cluster all Web services into top 20 categories (h = 20), we set the number of topics to 20 (T = 20) for all topic model-based methods, where each latent topic corresponds to a specific Web services domain. As for all Doc2Vec-based methods, the most important parameter is embedding size. With the higher number of features in a vector, the embedding model can capture more details.

D. Performance Evaluation

In this section, we first present the performance of all clustering methods mentioned above. Subsequently we report the effectiveness of the Doc2Vec-RK methods, compared with others baselines. The impacts of the balance parameter α is then investigated. Finally, we analyze the impacts of the service relationship information used in the proposed approach.

1) Clustering Performance Comparison

Table III presents the clustering performances of all baseline methods on top 20 categories. The bigger Precision, Recall and Purity, demonstrate that the clustering results are the better. Based on the results obtained, we have the following observations:

TABLE III. THE CLUSTERING PERFORMANCES OF DIFFERENT METHODS

Methods	Recall	Precision	Purity
TFIDF-K	0.0735	0.2692	0.1446
LDA	0.5206	0.4806	0.5351
Doc2Vec-K	0.5976	0.6017	0.5512
LDA-K	0.4695	0.4700	0.5289
LDA-PK	0.7566	0.7433	0.8568
Doc2Vec-RK	**0.7987**	**0.8122**	**0.8864**

- The Doc2Vec-RK is significantly superior to all other methods in various metrics by setting the same configurations as possible. the precision of Doc2Vec-RK has an improvement of 54.30% over the TFIDF-K, 33.19% over the LDA, 20.45% over the Doc2Vce-K and 6.89 % over the LDA-PK. The most important reason behind the results is that Doc2Vec-RK makes full use of both the services relationship information and services semantic information to improve the clustering accuracy. Despite its ability to incorporate complementary information, LDA-PK [20] only considers naive combination of services semantic information and some derived information. Our method can smoothly embed a services network into low-dimensional representations by modeling their structural proximities and relations. In addition, we can also conclude that Doc2Vec-PK performs better than the Doc2Vec-K. This again verifies the effectiveness of our model in incorporating services relationship information into the service clustering.
- The neural network based method Doc2Vec-K [21], shows a significant improvement of the clustering accuracy compared with the bag of word approaches (TFIDF-K) or topic models (LDA, LDA-K). This is because that these models do not consider the context information of a document (*i.e.*, order of words), which fails to capture implicit semantic correlations among Web services and also results in suboptimal representation. In addition, we observed that the Recall of TFIDF-K method is unexpectedly low. This is caused by the fact that in its clustering results the distributions of Web services are extremely unequal, which results in low Recall performance but high Precision performance in some categories containing very few Web services.

2) Hyper-parameter Investigation

d is a hyper-parameter that decides the dimensionality of implicit vector (S_i, R_i). It is recommended that with the dimension about 200, it would generate a good result of output. To study the impact of the dimensionality of implicit vector *d*, we set the dimensionality varying from 200 to 400 with a step value of 50. From Fig. 7, it can be seen that when increasing *d*, the values of Precision, Recall and Purity show an upward trend. More than 300 dimensions would still produce better performance. However, we also observe that too higher dimensions do not improve the clustering result much while the cost of time as well as the required memory for training would increase significantly. Therefore, we choose the dimensionality of the feature vectors as 300 in our method.

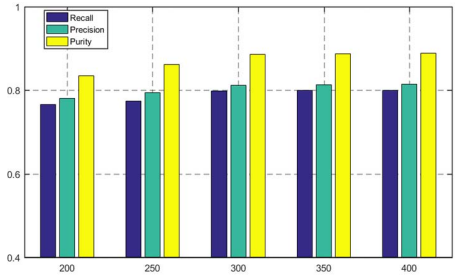


Figure 7. Performance *w.r.t* different *d*.

3) Impact of α

In this subsection, we investigate the effect of parameter *α*. The key parameter *α* in our algorithm balances the contributions of services semantic information and services relationship information. We represent the effect of changing *α* in Fig. 8 while varying it from 0.1 to 1. We fix $d = 300$ in these experiments. When $\alpha = 1$, the performance is totally determined by the services semantic information, so our method is downgraded to the traditional Doc2Vec-K model. From the Fig. 8, we can see that the performance of $\alpha = 0.5$ and $\alpha = 0.6$ are better than that of $\alpha = 1$ when both services semantic information and services relationship information have sufficient contributions. It demonstrates that both services semantic information and services relationship information are essential for services clustering. In general, the performances based on all metrics firstly show an upward trend and then decrease with *α*. A value of *α* around 0.7 gives relatively good performance, which demonstrates that service semantic information has a larger impact than services relationship information on our model. As a conclusion, our model could achieve relatively high performance by setting reasonable parameters.

4) Visualization of services relationship network

As we mentioned earlier, when the scale of the network grows larger and larger, the information it contains will be more instructive. It means that the vertex representations of the network have better quality. To demonstrate this, we construct services relationship networks with different level of accumulated usage data (links) by randomly removing different percentage of links among services. We keep the percentage of links in the services relationship network as {25%, 50%, 75%, 100%}. Then, a way of assessing the quality of the vertex representations *w.r.t* services relationship networks with different level of accumulated data is visualization. The vertex representation matrix was fed as features into t-SNE [26], which mapped all points into a 2D space, where the same category of services was highlighted

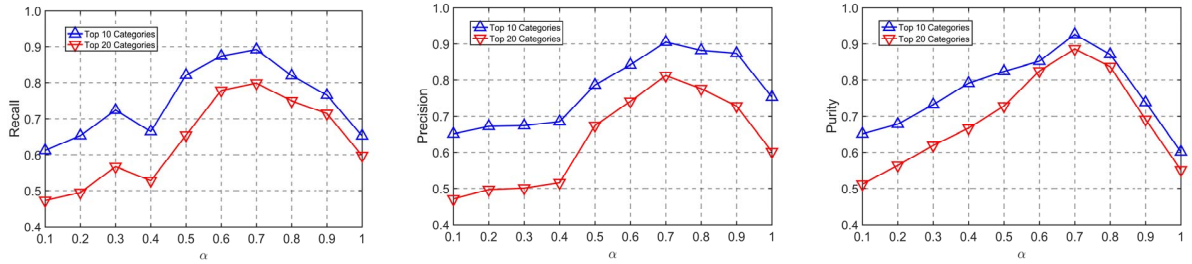
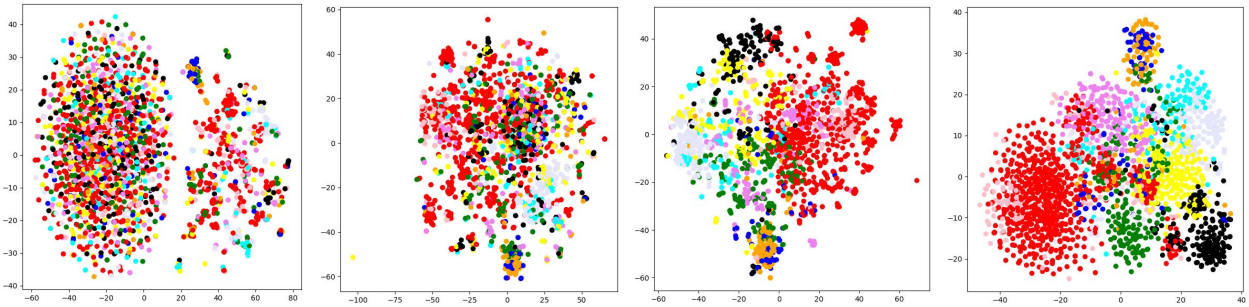


Figure 8. Impact of α



(a) Keep 25% of the links (b) Keep 50% of the links (c) Keep 75% of the links (d) Keep 100% of the links

Figure 9. Visualization of network embedding results. Vector representations serve as features to feed into t-SNE tool. Each point indicates a service. Color of a point indicates the category of the service

with the same color. Under the same setting, a cluster with clearer boundaries between different color groups indicates better representations. Fig. 9 shows the visualization of embedding vectors obtained from networks with different level of accumulated data using t-SNE toolkit under the same parameter configuration. From Fig. 9, for both (a) and (b), services from different categories are mixed with each other in the center of the figure. For (c), it forms 10 main clusters, which are better than those of (a) and (b). However, we can see the red points are still intermixed with some other points. Finally, we can observe that the visualization of (d) has 10 clusters with quite large margin between each other. Furthermore, the majority of the points of the same color are clustered together. The reason is that on the one hand, annotation relation (positive links) between services can bring together the same categories of services, on the other hand, composition relation (negative links) between services make a distinction between different categories of services. What is shown in (d) also offers good explanation about why the services relationship network can be informative for services clustering. Intuitively, this experiment demonstrates that the more accumulated usage data (a.k.a complicated network) we own, the better clustering quality we obtain.

V. RELATED WORK

Web service clustering technique plays a significant role in improving the quality of service discovery, management and etc. Lots of research work have been done on this direction now. They can roughly be categorized into three classes: non-functional-based Web services clustering [4], [10], function-based Web services clustering [3], [12] and their improved algorithms [18], [19], [20]. Non-functional-based clustering methods are based on QoS properties such as cost and reliability [10], [11], to which we do not pay attention since it is not relative to this paper.

Function-based Web services clustering methods focus on mining functional properties of Web services for clustering them into diverse service domains [3],]14]. In order to improve the clustering performance of web services, many enhanced algorithms have been proposed to mine the semantics from WSDL documents [17], [18]. For instance, Chen et al. [17] propose a method for Web service clustering by integrating both WSDL documents and tags based on the LDA model. However, most existing topic model-based methods elicit the latent topic information based on the WSDL documents [17], [18], which is difficult to obtain a well-performed topic model, especially when numbers of terms in description documents are limited. Thus, it may lead to unsatisfactory clustering accuracy as K-means algorithm heavily depends on the quality of latent topic vectors of services. Some auxiliary information is adopted to address the above problem [19, 20]. For example, to handle the word scarcity problem of service description documents, Shi et al. [19] propose to use the word cluster information to help eliciting better semantics based on the fact that Word2vec performs better than LDA in acquiring word embeddings. Considering human's trajectory of utilizing Web services, Shi et al [20] proposes a novel Web service clustering method by leveraging prior knowledge to enhance the clustering process in a semi-supervised way.

However, some obvious issues of the above approaches exist: 1) only utilizing services description documents; or 2) taking into consideration only a few related attributes and relationships among services and also being investigated in a

shallow level. Our approach proposed a thorough solution which incorporates the semantics both from the services description documents (services developers' side) and from the relationships between services (services users' side) shown in the network.

VI. CONCLUSION

This paper proposes a novel Web service clustering method that investigates services semantic information and services relationship information to assist the unsupervised clustering process. With services relationship information incorporated, the services clustering process is expected to be more accurate. Empirical comparisons with state-of-the-art models on a real-world dataset have demonstrated the validity and effectiveness of our model. We also investigate the impact of the parameter α and d, which are capable of exerting an influence on the clustering accuracy.

In future, a promising direction is to jointly project a services relationship network and services description documents into a unified embedding space by extracting their relations.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant 61572187, Grant 61872139, and Grant 61702181, in part by the Natural Science Foundation of Hunan Province under Grant 2017JJ2098, Grant 2018JJ2136, and Grant 2018JJ2139, and in part by the Educational Commission of Hunan Province of China under Grant 17C0642.

REFERENCES

[1] L.J. Zhang, J. Zhang and C. Hong, "Service-Oriented Architecture," *Services Computing*, 2007, pp. 89-113.

[2] S.Weerawarana, F. Curbera, F. Leymann, T. Storey and D.F. Ferguson,"Web services platform architecture: SOAP, WSDL, WS-policy, WS-addressing, WS-BPEL, WS-reliable messaging and more," Prentice Hall PTR, 2005.

[3] K. Elgazzar, A. Hassan, and P. Martin, "Clustering WSDL Documents to Bootstrap the Discovery of Web Services", in *2010 IEEE 17th International Conference on Web Services (ICWS)*. IEEE, 2010, pp. 147-154.

[4] T. Wen, G. Sheng, Y. Li, and Q. Guo, "Research on Web service discovery with semantics and clustering," in 2011 *IEEE 6th Joint International Information Technology and Artificial Intelligence Conference*. 2011, pp. 62-67.

[5] Y. Xia, P. Chen, L. Bao, M. Wang, and J. Yang, "A QoS-aware Web service selection algorithm based on clustering," in *2011 IEEE 18th International Conference on Web Services (ICWS)*. IEEE, 2011, pp.428-435.

[6] Z. Zhou, M. Sellami, W. Gaaloul, M. Barhamgi and B. Defude, "Data Providing Services Clustering and Management for Facilitating Service Discovery and Replacement," *IEEE Transactions on Automation Science and Engineering*, pp. 1131-1146, 2013 .

[7] C. Platzer, F. Rosenberg, and S. Dustdar,"Web service clustering using multidimensional angles as proximity measures," *ACM Transactions on Internet Technology (TOIT)*,vol. 9, no. 11,p. 26, 2009

[8] D. Skoutas, D. Sacharidis, A. Simitsis and T. Sellis, "Ranking and Clustering Web Services Using Multicriteria Dominance Relationships," *IEEE Transactions on Services Computing*, pp.163-177, 2010.

[9] B. Xia, Y. Fan, W. Tan, K. Huang, J. Zhang, and C. Wu, "Categoryaware API Clustering and Distributed Recommendation for Automatic Mashup Creation," *IEEE Transactions on Services Computing*, pp. 674-687, 2015.

[10] M. Zhang, X. Liu, R. Zhang, and H. Sun, "A Web service recommendation approach based on QoS prediction using fuzzy clustering," in *2012 IEEE 19th International Conference on Web Services (ICWS)*. IEEE, 2012, pp. 138-145.

[11] J. Zhu, Y. Kang, Z. Zheng, and M. R. Lyu, "A clustering-based QoS prediction approach for Web service recommendation". in *Proceedings of the IEEE 15th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops*. IEEE, 2012, pp. 93-98.

[12] M. Shi, J. Liu, D. Zhou, et al, "A Probabilistic Topic Model for Mashup Tag Recommendation". in *2016 IEEE 23th International Conference on Web Services (ICWS)*. IEEE, 2016, pp. 444-451.

[13] Q. Yu, H. Wang, and L. Chen," Learning Sparse Functional Factors for Largescale Service Clustering," in *2015 IEEE 22th International Conference on Web Services (ICWS)*. IEEE, 2015, pp. 201-208.

[14] Q. Xiao, B. Cao, X. Zhang, J. Liu, R. Hu and B. Li, "Web Services Clustering Based on HDP and SOM Neural Network," in *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI).* IEEE, 2018 pp. 397-404.

[15] D. Blei, A. Ng and M. Jordan,"Latent Dirichlet Allocation," *Journal of Machine Learning Research*. 2003, pp. 993-1022.

[16] Y.W. Teh, M. Jordan, M.J. Beal, et al. " Hierarchical Dirichlet Processes". *Journal of the American Statistical Association*, 2006,pp. 1566-1581.

[17] L. Chen, Y. Wang, Q. Yu, et al. "WT-LDA: User Tagging Augmented LDA for Web Service Clustering," in *International Conference on Service-Oriented Computing*. Springer, 2013, pp. 162-176.

[18] L. Chen, L. Hu, Z. Zheng, et al. "WTCluster: Utilizing Tags for Web Services Clustering". in *International Conference on Service-Oriented Computing*. Springer, 2011, pp. 204-218.

[19] M. Shi, J. Liu, D, Zhou, et al. "WE-LDA: A Word Embeddings Augmented LDA Model for Web Services Clustering," in *2017 IEEE 24th International Conference on Web Services (ICWS)*. IEEE, 2017, pp. 9-16.

[20] M. Shi, J. Liu, B. Cao, Y. Wen and X. Zhang, "A Prior Knowledge Based Approach to Improving Accuracy of Web Services Clustering," in *2018 IEEE 15th International Conference on Services Computing (SCC)*. IEEE, 2018, pp. 1-8.

[21] Q.V. Le, T. Mikolov, "Distributed Representations of Sentences and Documents," in *Proceedings of the 31th International Conference on International Conference on Machine Learning*. ACM, 2014, pp. 1188-1196.

[22] D. Zhang, J. Yin, X. Zhu and C. Zhang, "Network Representation Learning: A Survey," *IEEE Transactions on Big Data*. pp.1-25,2018.

[23] S. Wang, J. Tang, C. Aggarwal, Y. Chang, H. Liu, "Signed network embedding in social media," in *Proceedings of the 17th SIAM International Conference on Data Mining*. SDM, 2017, pp. 327-335.

[24] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 701–710.

[25] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 855–864.

[26] L. Maaten and G. Hinton,"Visualizing data using t-sne," *Journal of Machine Learning Research*, pp. 2579–2605, 2008.