Computing and Informatics, Vol. 44, 2025, 853–881, doi: 10.31577/cai_2025_4_853

SEMANTIC ENHANCEMENT AND HETEROGENEOUS CORRELATION GUIDED WEB SERVICE CLUSTERING

Haoquan Qi

School of Computer Science and Technology
Donghua University
Shanghai, China
e-mail: 1249141@mail.dhu.edu.cn

Bing Wang

Department of Information Engineering
Shandong Water Conservancy Vocational College
Rizhao, China
e-mail: wangbing6666@vip.163.com

Qiang Hu

College of Information Science and Technology
Qingdao University of Science and Technology
Qingdao, China
e-mail: huqiang200280@163.com

Pengwei Wang*

School of Computer Science and Technology
Donghua University
Shanghai, China
e-mail: wangpengwei@dhu.edu.cn

* Corresponding author

Abstract. Service description suffers from short texts and contains few repeated words, which brings challenges to generate high-quality service function vector (SFV) in service clustering. Some works introduce service association to improve service clustering quality. However, they simply introduce single associations, such as tag associations or collaboration association. Single service association can only improve the clustering quality from one perspective of positive or negative categorical relevance. In this study, we propose semantic enhancement and heterogeneous correlation guided Web Service Clustering. A high-performance contrastive learning framework is employed to generate SFVs. Meanwhile, we propose a method for the semantic enhancement of SFVs by obtaining twin service descriptions through verb substitution. A heterogeneous association is established based on tag association and collaboration association. It quantitatively enhances the clustering quality from both positive and negative categorical relevance. Experiments show that the proposed method outperforms popular semantic enhancement ways in generating high-quality SFVs. The heterogeneous association can significantly improve service clustering quality compared to single tag association or collaborative association. The clustering quality obtained by our method is improved by 13.7 %, 9 %, 6.8 %, 6.1 %, and 5.5 % on average over the state-of-the-art service clustering methods in terms of DBI, SC, AMI, NMI, and Purity.

Keywords: Web service, service clustering, contrastive learning, heterogeneous association

Mathematics Subject Classification 2010: 68-T20

1 INTRODUCTION

Web service is a popular way to organize various types of service APIs on the Internet. It is a network program module with specific business functions encapsulated by standardized protocols and interfaces. Currently, more and more software systems are deployed based on the service-oriented architecture (SOA) [1]. By invoking and integrating Web services, software developers can realize agile development and rapid iteration of software business systems with SOA.

There are many service registration platforms on the Internet, such as ProgrammableWeb [2] and RapidAPI [3]. Numerous Web services have been offered on these platforms. For example, service providers have registered nearly 27 000 Web services and 8 000 Mashup services, covering more than 500 application types in the ProgrammableWeb. A tremendous number of Web services provide the users with abundant opportunities to choose appropriate services for building new business systems. However, quickly finding the services that meet the needs from those massive Web services has become a challenge [4, 5].

Service clustering aggregates Web services with similar functions and classifies them into different service groups. It can reduce the search space by merging

Web services with similar functional features. Thus, service clustering is frequently used to improve the efficiency of service discovery [6, 7]. Service providers generally use natural language to describe the functions of Web services. Topic models or neural networks are usually employed to generate SFVs based on these service descriptions. Then, service clustering is performed by evaluating the similarity of SFVs [8]. The service description is generally a short text of about 100 words. It has a high proportion of nouns and few repeated words [9]. So the topic features in SFVs are sparse, and their differentiation degree is insignificant. However, the verbs with a relatively low proportion in service description usually represent the key business operations of Web services. Therefore, if the functional feature density corresponding to the verbs in service description is increased, the quality of the SFV may be improved. Thus, one of the motivations of our study is to build a semantically enhanced generation method for SFVs by increasing the feature density of verbs in service descriptions, so as to improve the quality of service clustering.

In recent years, service associations have gradually attracted concerns in service clustering studies [10]. Tag association and collaboration association are often used to improve the service clustering quality. Two Web services are considered to have tag associations if they have one or more identical service tags [11]. Services with tag association imply similar functions and are more likely to be classified into one group. So, tag associations can improve service clustering quality from the positive categorical relevance of service aggregation.

Two Web services are considered to a have collaboration association if they occur together in a business scenario (for example, a Mashup service or a service composition). It is generally believed that two services with a collaboration association are functionally complementary. Therefore, Web services with a collaboration association are more likely to be classified into different groups in the service clustering. So, collaboration associations can produce a negative categorical relevance of service aggregation in service clustering.

Integrating both tag associations and collaboration associations into service clustering can improve clustering quality by considering both the positive and negative relevance of service aggregation. However, determining how to quantitatively measure the impact of these two associations on service similarity remains a challenging problem. To the best of our knowledge, no research work is addressing the above issue. Another motivation of our study is to combine tag association and collaboration associations to form a new service association, which can further enhance the service clustering quality in categorical relevance.

Based on the above two motivations, we propose semantic enhancement and heterogeneous correlation guided Web service clustering. The main contributions are as follows:

1. A method to obtain high-quality SFVs using contrastive learning and semantic enhancement is proposed. We construct twin service descriptions by synonym substitution to increase the semantic features of verbs representing service opera-

tions, and generate high-quality SFVs under the framework of simple contrastive learning of sentence embeddings (SimCSE).

2. A service heterogeneous association graph is proposed to model tag association and collaboration association between Web services simultaneously. Service clustering quality is enhanced from both positive and negative categorical relevance by quantizing the influence of heterogeneous association on service similarity.
3. We evaluate the proposed method through extensive experiments, which show that semantic enhancement and heterogeneous association can guide Web service clustering. Our proposed service clustering method outperforms the state-of-the-art clustering methods.

The remainder of this paper is organized as follows: Section 2 introduces related work on service clustering. Section 3 presents the preliminary knowledge about Web service and service associations. Section 4 details our service clustering method. Section 5 evaluates the performance of the proposed method. Finally, Section 6 concludes the paper and outlines future work.

2 RELATED WORK

We briefly review three lines of research closely related to our work, including topic model-based service clustering, neural network-based service clustering and service clustering with service association [12].

2.1 Topic Model-Based Service Clustering

A variety of topic models are exploited to generate SFVs in service clustering. For example, Shen et al. [13] proposed a variant model of LDA with a probability incremental correction factor (PICF-LDA) to produce service representation vectors for Web APIs. PICF-LDA has outperformed the existing variant LDA models in service clustering. Yang and He [14] used the Biterm topic model and Gibbs sampling to achieve high-quality feature extraction for Web services. Experimental results show that the proposed method can improve the clustering effect of Web services. The hierarchical Dirichlet processes model was used to mine the topic information on API service documents by Cao et al. [15], then the SOM neural network was employed to cluster API services into various clusters with similar topics and functions.

Agarwal et al. [16] generated SFVs by Gibbs sampling algorithm for the Dirichlet Multinomial Mixture (GSDMM). Experiments show that the quality of SFVs generated by GSDMM is significantly higher than that of other topic models. Furthermore, genetic algorithm can also improve the quality of service clustering [17]. In the latest research work, Agarwal et al. [18] combined the genetic algorithm with GSDMM, further improving service clustering quality.

Although many improvements have been made to the topic models, the quality of SFVs is not significantly enhanced. The main reason is that service descriptions

suffer from data sparsity and lack of repeated words. Topic models employ probability statistics on words in the text to obtain topic features. Therefore, it is not conducive to the topic model to extract the topic features in service descriptions.

2.2 Neural Network-Based Service Clustering

In recent years, a series of neural network models have been applied to extract the functional features of Web services, which greatly improves the generation quality of SFVs. For example, Ye et al. [19] proposed a WSC-GCN classification model based on a graph neural network and constructed an undirected graph by taking word-document and word-word relations as edges and word-document as nodes. The value of TF-IDF was used as the weight of the side and put into the GCN graph neural network to obtain the document vector for clustering. Tang et al. [20] proposed a novel deep neural network with the co-attentive representation learning mechanism for effectively classifying services by learning interdependent characteristics of Web services. The classification quality of this method was better than that of CNN, LSTM, Recurrent-CNN, C-LSTM and BLSTM.

Kang et al. [21] proposed a Web service classification approach with a topical attention-based Bidirectional Long Short-Term Memory Network. The enhanced Web service feature representation is used as the input of a softmax neural network layer to perform the classification prediction for Web services. Zhu et al. [22] proposed a deep manufacturing cloud service clustering model using pseudo-labels (DSCPL). DSCPL combined graph topology and node features to cluster nodes with similar attributes. An auxiliary target distribution was presented to realize the self-learning mechanism in order to adapt to the clustering task. Zou et al. [23] presented a novel heuristics-based framework DeepWSC for web service clustering. It integrated deep semantic features extracted from service descriptions by an improved recurrent convolutional neural network and service composability features obtained from service invocation relationships by a signed graph convolutional network. Together, these elements generated integrated implicit features for web service clustering. Ping et al. [24] constructed the networks for describing text and tags, respectively, and merged the two networks to form a web service network. They proposed an efficient document weight and tag weight-LDA model to generate SFVs that can perform high-quality service clustering.

Compared with topic model, the neural network model can make full use of the context information to effectively learn the semantic and sentence structure features of words in the service description. Therefore, neural network model is more suitable for extracting service functional features. Using neural network models to obtain SFVs can construct higher quality service clustering.

2.3 Service Clustering with Service Association

Web services can form a network with some kind of association, such as service tags, collaborations, or service providers. The association intensity between these services

can be calculated by vectorizing Web services in a certain association network. Service clustering quality will be improved once the service association intensity is introduced into the clustering process. Cao et al. [25] pointed out that rich network relations inherently reflect either positive or negative categorical relevance between services, which can strongly supplement service semantics in characterizing the functional affinities between services. They utilized the tag-sharing relation network to enhance service clustering, which yields an improvement of 6.89 % compared to the state-of-the-art methods.

Hu et al. [26] proposed a service collaboration graph to model collaboration associations between Web services. They employed Node2vec to vectorize the nodes in service collaboration graph and integrated the collaboration similarity into service clustering. Experiments show that the quality of service clustering is improved by about 10 % after the introduction of service collaboration. Kang et al. [27] constructed an association network according to the service structure relationship. They integrated the tag association and collaboration association into service clustering. Experimental results demonstrate that the proposed method yields an improvement of 4.78 % in precision and 5.4 % in recall over the state-of-the-art method.

From the above work, we can see that generating high-quality SFVs and integrating service associations can heighten the quality of service clustering. Aiming to provide high-quality service clustering, we guide the Web service clustering through semantic enhancement and heterogeneous association. Our study employs a high-performance contrastive learning framework (SimCSE) to generate SFVs. Meanwhile, we propose a method for the semantic enhancement of SFVs by generating twin service descriptions through verb substitution. Moreover, a heterogeneous association combining tag association and collaboration association is introduced to enhance the quality of service clustering further.

3 PRELIMINARY KNOWLEDGE

In this section, we provide formal definitions related to Web service and service associations, and outline the problem to be solved.

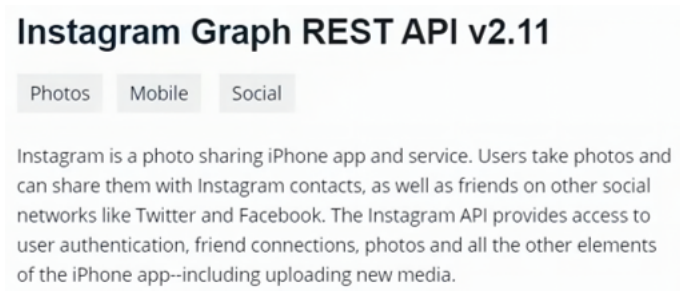


Figure 1. An example of a Web service

Web services in this study refer to the popular Web APIs that use natural language to describe their service functions. Figure 1 provides an example of a Web service. Typically, the following information can be found for Web services in the service registration platforms: the service name, the service tags, and the service description in the form of short text.

The service name of the Web service in Figure 1 is Instagram Graph. The service tags are photos, mobile, and social, while the text below the service tags is its service description. The formal definition of Web service can be referred to as Definition 1.

Definition 1 (Web service)**.** A Web service is defined as a 4-tuple $s = (id, n, T, d)$, where id is the service identification, n is the service name, T is the set of service tags, and d is the service description.

Web services do not exist in isolation. For example, different Web services may be followed by the same user. Multiple Web services with similar functionality may belong to the same service provider. In addition, Web services may need to interact with each other to accomplish complex tasks. In the above scenarios, there is a certain type of association between Web services. Service associations imply the categorical relevance between Web services. For example, Web services with similar partners prefer to be classified into one class. Therefore, service associations can be employed to improve service clustering quality. The definitions of tag association, collaboration association and heterogeneous association graph used in this study are presented below.

Definition 2 (Tag association)**.** Web services s_i and s_j are called tag association if there exists a service tag t such that $t \in s_i.T \cap s_j.T$, denoted as $s_i \sim s_j$.

Definition 3 (Collaboration association)**.** Web service s_i and s_j is called collaboration association if they participate in the same service composition, denoted as $s_i \leftrightarrow s_j$.

Definition 4 (Heterogeneous association graph)**.** A heterogeneous association graph is an undirected weighted graph HAG = (V, E, W) if the following hold:

1. $V = \{v_1, v_2, \ldots, v_n\}$ is a set of service nodes while the node v_i represents Web service s_i.
2. $E = \{Ef, Ec\}$, Ef and Ec are the edge set of tag association and collaboration association, respectively.
   (a) $ef = (v_i, v_j) \in Ef$ once s_i and s_j satisfy $s_i \sim s_j$.
   (b) $ec = (v_i, v_j) \in Ec$ once s_i and s_j satisfy $s_i \leftrightarrow s_j$.
3. $W = \{Wef, Wec\}$, Wef and Wec are the edge weight set of tag association and collaboration association, respectively.
   (a) $\forall ef = (v_i, v_j) \in Ef$, $wef_{ij} = Nt(s_i, s_j)$;
   (b) $\forall ec = (v_i, v_j) \in Ec$, $wec_{ij} = Ns(s_i, s_j)$;

$Nt(s_i, s_j)$ is the edge weight of the tag association between s_i and s_j. The value of $Nt(s_i, s_j)$ is the number of common tags in s_i and s_j. Similarly, $Ns(s_i, s_j)$ is the edge weight of the collaboration association. Its value is the number of service compositions in which s_i and s_j jointly participate.

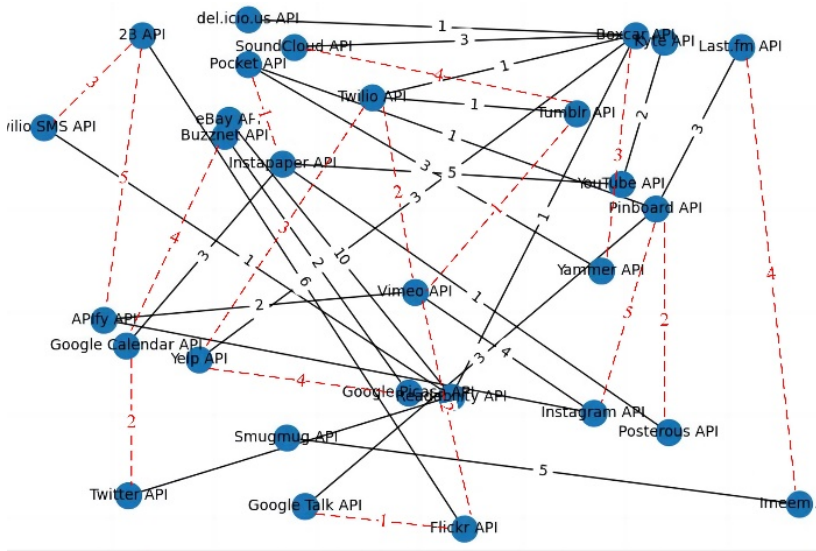


Figure 2. A fragment of HAG

Let $S = \{s_1, s_2, \ldots, s_n\}$ be a group of Web services. The goal of this study is to generate a series of service clusters, denoted as $C = \{c_1, c_2, \ldots, c_m\}$, for the Web services in S. We hope that the proposed method can make the service similarity within each service cluster in C as high as possible, and the service similarity among different service clusters as low as possible.

The task of the proposed method is to extract the high-quality functional semantic features and combine service associations to facilitate clustering the Web services. We provide a way to enhance the quality of semantic feature extraction of SFVs. Heterogeneous graph representation learning is exploited to achieve association embedding between Web services. Thus, we can guide Web service clustering through semantic enhancement and heterogeneous association.

4 METHODOLOGY

4.1 Overview of the Proposed Approach

The pipeline of this study is shown in Figure 3, which consists of four components. The first component is the data crawling module. We crawl Web services and service

compositions from some service registration platforms. A service composition may be a business scenario composed of several Web services, or it may be some composite services, such as Mashup services. In this module, we can obtain service tags, service descriptions and service collaborations for Web services.

The second component is the semantic enhancement module. Twin service descriptions are bred by substituting the verbs in original service descriptions. They are employed to enhance functional semantic features in SFV. Moreover, contrast learning is also exploited to generate high-quality SFVs.

The service association module is the third component. Tag association and collaboration association are integrated into HAG. The improved random walk strategy and GATNE (General Attributed Multiplex Heterogeneous Network Embedding) model are employed to achieve node embedding for the HAG. Service association vectors (SAVs), which can further enhance service clustering quality, can be obtained in this module.

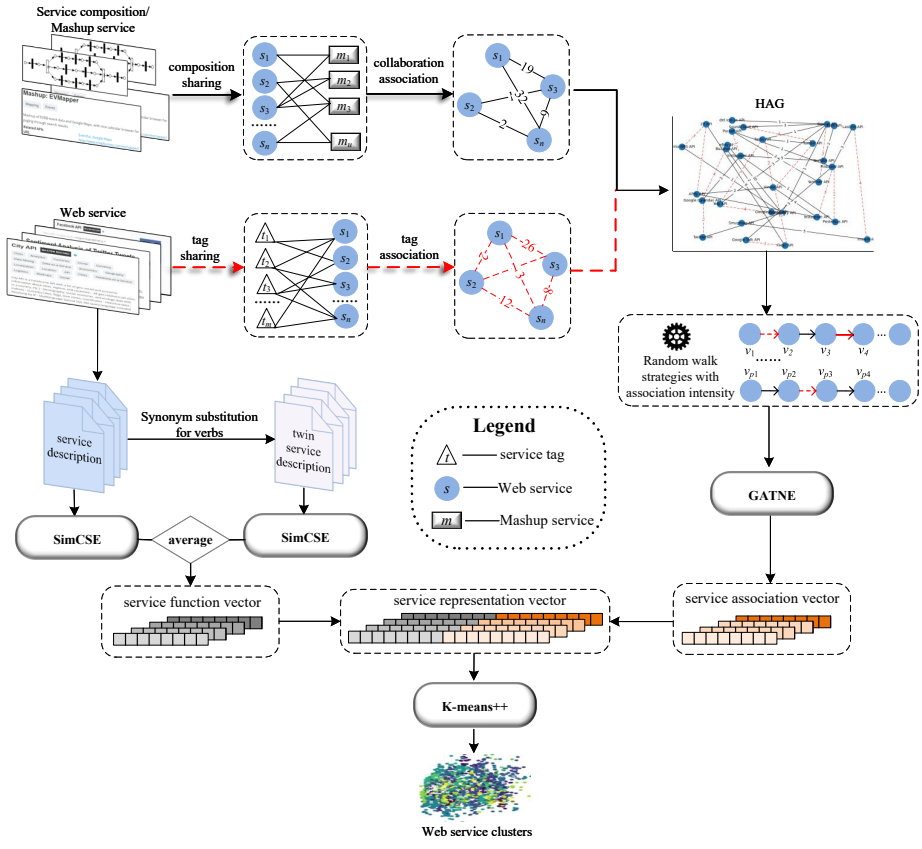


Figure 3. Pipeline of the proposed method

The fourth component is designed for service clustering. SFV and SAV are concatenated as the service representation vector (SRV) in this module. Then, K-means++ algorithm is adopted to achieve service clustering based on SRVs. Similar to other service clustering studies, we preprocess the crawled service descriptions by tokenization, stop word removal, and stemming. The following sections focus on the latter three modules in the proposed method.

4.2 SFVs with Semantic Enhancement

Compared with topic models, the neural network models have better feature capture abilities [28]. BERT and its variant models are frequently employed to generate SFVs. However, BERT suffers from anisotropy, which affect the accuracy of similarity calculation of the generated vectors. The SimCSE proposed by Gao et al. [29] can effectively alleviate the anisotropy.

Let $D = \{d_1, d_2, \ldots, d_n\}$ be a set of the textual corpus, where d_i is a text sample. $h_i = f_\theta(d_i)$ is denoted as the text vector of d_i. Here f_θ may be the BERT or its variant model with fine-tuned parameters. In the SimCSE, d_i is fed to the encoder twice to obtain two embeddings h_i^z and $h_i^{z'}$ with different random dropout masks z and z.

$$h_i^z = f_\theta(d_i, z), \tag{1}$$

$$h_i^{z'} = f_\theta(d_i, z'). \tag{2}$$

Let h_i and h_i^+ denote the representation of h_i^z and $h_i^{z'}$. (h_i, h_i^+) is the positive pair for the d_i. Then h_i and h_j^+ obtained by the text sample d_j in the same batch through dropout mask z is employed to generate negative pairs. The loss is formalized as:

$$\rho = -\log \frac{e^{sim\left(h_i, h_i^+\right)/\tau}}{\sum_{j=1}^{N} e^{sim\left(h_i, h_j^+\right)/\tau}}. \tag{3}$$

Here, τ is a temperature hyperparameter. $sim(h_1, h_2)$ is the cosine similarity of h_1 and h_2. It optimizes contrastive loss by increasing the similarity between positive pairs and reducing the similarity between negative pairs. In this way, the singular value distribution of sentence embedding space can be uniform and the consistency of the whole representation space can be improved. The vector h_i of d_i can be obtained once the loss converges.

SimCSE adopts the contrastive learning way to produce superior sentence embedding. We introduce SimCSE to generate high-quality SFVs in this study. Meanwhile, we propose a method for the semantic enhancement of SFVs by obtaining twin service descriptions through verb substitution. The verbs in the service description are usually used to represent the operations performed by Web service. Therefore, if the semantic features of verbs in the service description is enhanced, it will be more likely to generate high-quality SFVs.

$D = \{s_1.d, s_2.d, \ldots, s_n.d\}$ is used to denote the set of service descriptions, where $s_i.d = \{w_1, w_2, \ldots, w_n\}$ is the description text of Web service s_i. The StanfordNLP tool is used to tag the part-of-speech of words in $s_i.d$. We assume that a total of q verbs exist in $s_i.d$ and these verbs are denoted as $\{w_{j-p+1}, \ldots, w_{j-p+k}, \ldots, w_{j-p+q}\}$.

We can use WordNet to find synonyms for a word. Let $Sy = \{w_{j-p+k,1}, w_{j-p+k,2}, \ldots, w_{j-p+k,t_j}\}$ be the synset of w_{j-p+k}. Word2Vec is employed to generate the word embedding for all the words in Sy and w_{j-p+k}. We use the angle cosine method to calculate the semantic similarity of words in Sy and w_{j-p+k} based on their word embedding. The word with the largest semantic similarity is used as the substitute synonym for w_{j-p+k}. The service description generated after replacing all verbs in $m_i.d$ with their synonyms is denoted as $s_i.d$, which is called the twin sample of $s_i.d$. D' is used represent the set of twin samples for D.

Algorithm 1 SFV-SE-SimCSE

Input: the set of Web services S;
Output: T_{sfv} (the set of SFVs for S);
(1) $D = \{s.d | s \in S\}$;
(2) $D' = \emptyset$; $T_{sfv} = \emptyset$;
(3) ***for each*** service s in S
(4) $d' = \emptyset$;
(5) ***for each*** word w in $s.d$
(6) ***if*** $(pos(w) == verb)$ then;
(7) $w' = \{sw | maxSemSim(sw, w) \cap swsynsets(w)\}$;
(8) ***else*** $w' = w$;
(9) $d' = d' \cup \{w\}$;
(10) ***end if***;
(11) ***end for***;
(12) $D = D' \cup \{d'\}$;
(13) ***end for***;
(14) ***for each*** service s in S
(15) $h = SimCSE(s.d, D)$;
(16) $h' = SimCSE(s.d', D')$;
(17) $sfv(s) = (h + h')/2$;
(18) $T_{sfv} = T_{sfv} \cup \{sfv(s)\}$;
(19) ***end for***;
(20) ***return*** T_{sfv};

Algorithm 1 provides the method to generate SFV in this study. In line (1) and line (2), we first build the set of service descriptions D and initialize two empty sets. One is the set of twin service descriptions D, and the other is the set of SFVs. The codes in line (3) to line (13) are used to generate a set of twin service descriptions D. For each service description d in D, a twin service description d' is constructed. We detect the part of speech of each word in $s.d$, replacing the word with a verb by its synonym. If the word w is a verb, the synonyms of w are obtained from WordNet.

The word w and its synonyms are transformed into word vectors by Word2Vec. Then we can calculate the similarity between the word w and its synonyms based on these word vectors. The synonym with the highest similarity is selected as the substituted word in the twin service description. In the line (7), maxSemSim and synsets denote maximum semantic similarity and synonym set, respectively. Non-verbal words will be placed directly in the twin service description.

The twin service descriptions are exploited in line (14) to line (19) to achieve semantic enhancement in generating SFVs. For each Web service s, we utilize the SimCSE to produce SFVs h and h' for the original service description d and the twin service description d', respectively. The mean value of h and h' is taken as the final SFV of Web service s.

4.3 SAVs with the Heterogeneous Association

We model tag association and collaboration association as heterogeneous association. To evaluate the heterogeneous association intensity between two Web services, we need to generate SAVs for the nodes in the HAG. An improved GATNE and optimal sampling strategy are employed to obtain SAVs in this section.

4.3.1 Generation of SAVs by the Improve GATNE

GATNE is a popular embedding model for heterogeneous graphs [30]. The node embedding on each edge type r in the GATNE model includes base embedding and edge embedding. The k^{th} level edge embedding $u_{i,r}^{(k)}$ of node v_i on edge type r is aggregated from its neighbors' edge embedding as:

$$u_{i,r}^{(k)} = \text{aggregator}\left(\left\{u_{j,r}^{(k-1)}, \forall v_j \in N_{i,r}\right\}\right), \tag{4}$$

where $N_{i,r}$ is the set of v_i associated nodes on the edges of type r. The initial edge embedding $u_i^{(0)}$, for each node v_i and the edge type r is randomly initialized. Formula (5) is used to implement the max-pooling aggregation on $u_{j,r}^{(k)}$, and $\varphi(x)$ is the ReLu activation function.

$$u_{i,r}^{(k)} = \max\left(\left\{\varphi\left(\hat{Q}_{pool}^{(k)} u_{j,r}^{(k-1)} + \hat{b}_{pool}^{(k)}\right), \forall v_j \in N_{i,r}\right\}\right). \tag{5}$$

Finally, all the edge embedding of node v_i are concatenated. Given that there are m edges of type r associated with v_i and the dimension of edge embedding is s, the concatenated edge embedding is characterized by $U_{i,r} = (u_{i,1}, u_{i,2}, \ldots, u_{i,m}), U_i \in \mathrm{R}^{[s\times m]}$. It employs a self-attention mechanism [31] to compute the coefficients $a_{i,r} \in \mathrm{R}^m$ of the linear combination of vectors in $U_{i,r}$ as Formula (6).

$$a_{i,r} = \text{softmax}\left(w_r^T \tanh\left(W_r U_{i,r}\right)\right)^T. \tag{6}$$

Here, w_r and W_r are trainable parameters for edge type r with size da and $da \times s$ respectively and the superscript T denotes the transposition of the vector or the matrix.

However, GATNE cannot deal with the edge weights of heterogeneous graphs in generating the edge embedding. It treats all adjacent edges of each node with equal weight. The edge weight indicates the association closeness between Web services. Ignoring edge weights makes GATNE difficult to distinguish the importance of nodes in different adjacent neighborhoods, which reduces the aggregation quality of service associations.

To address the above problem, we present a bias term with edge weights to correct the attention coefficient in aggregating edge embedding features. The bias term can help GATNE to distinguish the node importance in different neighborhoods. The bias term of node v_i and v_j under edge type r is $b_{i,j}^k$.

$$b_{i,j}^k = \sigma\left(W_w^k u_{i,j}^k\right). \tag{7}$$

The optimized attention coefficient is shown in formula (8), while the overall embedding of node v_i for edge type r is given in formula (9).

$$a_{i,r} = \text{softmax}\left(w_r^T \tanh\left(W_r U_i\right) + \sum_{j=1}^{m} w_{i,j} b_{i,j}^k\right)^T m, \tag{8}$$

$$\nu_{i,r} = b_i + \alpha_r M_r^T U_i a_{i,r}. \tag{9}$$

In formula (9), b_i is the base embedding of node v_i, α_r is a hyper-parameter, which is the attention coefficient assigned to different types of edges, and Mr is a trainable transformation matrix.

4.3.2 Optimization of SAVs

Metapath2vec++ is suitable to optimize the node embedding for GATNE [32, 33]. It randomly samples nodes with an equal probability according to the type of heterogeneous edges. When sampling in the HAG of Web API, the nodes with the strongest association cannot be preferentially included in the path sequence, which affects the determination of association compatibility. To this end, we design a random walk strategy oriented to service association intensity. How to calculate the service association intensity can refer to formulas (10), (11) and (12).

$$Ief_{ij} = \frac{\sum_{t \in s_i.T \cap s_j.T} Ns\left(t\right)}{\sum_{t \in s_i.T} Ns\left(t\right) + \sum_{t \in s_j.T} Ns\left(t\right)}, \tag{10}$$

$$Iec_{ij} = \frac{Nsp\left(s_i\right) + Nsp\left(s_j\right)}{Nsp(s \in WS\_sp(s_i) \cup s \in WS\_sp(s_j))}, \tag{11}$$

$$Cor_d(v_i, v_j) = Ief_{ij} * Wef_{ij} - Iec_{ij} * Wec_{ij}. \tag{12}$$

For a pair of nodes (v_i, v_j) in the HAG, the service association intensity of v_i and v_j is expressed as the difference between tag association intensity and collaboration association intensity. The first half of the formula (12) is the tag association intensity. It is calculated by the product of the importance degree of the tag association edge Ief_{ij} and its weight Wef_{ij}. The numerator of Ief_{ij} is the total number of common tags of s_i and s_j invoked by all Web services. The number of all tags of s_i and s_j invoked by the Web services is placed in the denominator of Ief_{ij}.

The second half of the formula (12) is the collaboration association intensity. It is calculated by the product of the importance degree of the collaboration association edge Iec_{ij} and its weight Wec_{ij}. Iec_{ij} is represented as the ratio of $Nsp(s_i)+Nsp(s_j)$ and $Nsp(s \in WS\_sp(s_i) \cup s \in WS\_sp(s_j))$. Here, $Nsp(s)$ denotes the number of service compositions containing Web service s. $WS\_sp(s)$ is employed to represent the set of Web services contained in the service composition containing Web service s. Then, $Nsp(s_i) + Nsp(s_j))$ is the total number of service compositions containing s_i or s_j. For $Nsp(s \in WS\_sp(s_i) \cup s \in WS\_sp(s_j))$, we first obtain all the Web services included in the service compositions containing s_i or s_j, i.e, $s \in WS\_sp(s_i) \cup s \in WS\_sp(s_j)$. Then, we count the number of service compositions that the above Web services participate in as the final value. The softmax function is used to normalize the service association intensity as formula (13).

$$NCor\_d\,(s_i, s_j) = \frac{e^{Cor\_d(s_i,s_j)}}{\sum_{s_j \in N_{l+1}(v_{i,r})} e^{Cor\_d(s_i,s_j)}}. \tag{13}$$

For the sampling sequence $L = v_1 \to v_2 \to \ldots v_l \to v_{l+1} \to v_q$, if the type of walking edge at step l-1 is r, then the transition probability integrated with service association intensity from v_i to v_j at step l is:

$$p\left(v_j \middle| v_i, T\right) = \begin{cases} \frac{NCor\_d(s_i,s_j)}{|N_{t+1}(v_{i,r})|}, & (v_i, v_j) \in E, v_j \in N_{t+1}(v_{i,r}), \\ 0, & (v_i, v_j) \in E, v_j \notin N_{t+1}(v_{i,r}), \\ 0, & (v_i, v_j) \notin E. \end{cases} \tag{14}$$

If the node v_i is wandering from an edge of r type, the next visiting node v_j should be in the associated node set that is not with the edge type r. Then, the node with the highest probability $p(v_j|v_{i,L})$ is selected to be visited. Let v_k be the previously visited node of v_i and $v_J = \{v_{j1}, v_{j2}, \ldots, v_{jm}\}$ be the next visiting node-set. The rules for selecting the next node are as follows:

1. If $v_i \leftrightarrow v_k$, $\exists v_{Jf} \in v_J$ and $\forall v \in v_{Jf}$ s.t. $v \sim v_i$, the next visiting node is $v_j = \{v|v \in v_{Jf} \cap max(p(v|v_i, T)\}$.
2. If $v_i \sim v_k$, $\exists v_{Jc} \in v_J$, and $\forall v \in v_{Jc}$ s.t. $v \leftrightarrow v_i$, the next visiting node is $v_j = \{v|v \in v_{Jc} \cap max(p(v|v_i, T)\}$.

Here, $N_{l+1}(v_i, r)$ denotes the set of associated nodes of v_i whose edge type is not r. For a random walk sequence $P = (v_{p1}, \ldots, v_{pl}, v_{pl+1}, \ldots, v_{pq})$, the context of

v_{pl} is defined as $C = \{v_{pk}|v_{pk} \in P, |k-l| \leq c, l \neq k\}$, where c is the radius of the window size. Then, the following likelihood function is minimized for node v_i and its sequence context C.

$$-\log P_\theta\left(\{v_j\,|v_j \in C\}|\, v_i\right) = \sum_{v_j \in C} -\log P_\theta\left(v_j\,|v_i\right). \tag{15}$$

Next, the softmax function is used to normalize the context appearance probability of node v_j as formula (16).

$$P_\theta\left(v_j\middle|v_i\right) = \frac{exp\left(C^T \cdot v_{i,r}\right)}{\sum_{k \in V_l} exp\left(c_k^T \cdot v_{i,r}\right)}. \tag{16}$$

Here, $v_j \in V_l$, c_k is the contextual embedding of node v_k, and $v_{i,r}$ is the overall embedding of node v_i to the edge type r. Finally, the optimization objective function is constructed for each pair of nodes by negative sampling function.

$$E = -\log\sigma\left(c_j^T \cdot v_{i,r}\right) - \sum_{ne=1}^{NE} E_{v_k \sim P_l(v)}\left[\log\sigma(-c_k^T \cdot v_{i,r})\right], \tag{17}$$

where $\sigma(x)$ is the sigmoid activation function, NE is the number of negative samples obtained by negative sampling for each positive sample, and v_k is randomly selected from the noise distribution $P_l(v)$ on the node set V_l corresponding to the node v_j. When the objective function converged, the final SAVs were generated for the nodes in the HAG of Web APIs.

In summary, we improve the existing methods from two perspectives: the sequential sampling method of Metapath2vec++ and the node feature aggregation of GATNE. The optimized method can effectively improve the generation quality of SAVs. We named the improved method as MG-HAG. Algorithm 2 presents the steps to obtain the SAVs based on the MG-HAG.

We provide two sets for the input of Algorithm 2. One is the set of Web services S, the other is the set of service compositions SP. In the line (1), the Web services in S are used to initialize the nodes in graph G. Line (2) to line (7) of the algorithm establish tag association edges for graph G by detecting whether there is a tag sharing between any two services. Similarly, the collaboration association edges of graph G are generated in line (8) to line (12). By traversing each composition scenario sp in SP, collaboration association edges are established between each pair of services in the sp. A new edge is created for the edge that does not exist while the edge weight is updated for the already existing edge. After generating the heterogeneous association graph G, we invoke the MG-HAG to generate SAVs for all the Web services in line (14). The last line of Algorithm 2 is used to return the set of service association vectors T_{sav}.

Algorithm 2 MG-HAG

Input: the set of Web services S, the set of service compositions SP;
Output: the set of service association vectors T_{sav} for S;

(1) $G \leftarrow v : s \in S$;
(2) ***for each*** $s_i, s_j \in S$;
(3) ***if*** $s_i \sim s_j$ ***then***
(4) $G.Ef = G.Ef \cup (v_i, v_j)$;
(5) $w_{ij} = w_{ij} + 1$;
(6) ***end if***
(7) ***end for***
(8) ***for*** $\forall sp \in SP$
(9) ***if*** $\exists s_i, s_j \in sp \cap s_i \leftrightarrow s_j$ ***then***
(10) $G.Ec = G.Ec \cup (v_i, v_j)$;
(11) $w_{ij} = w_{ij} + 1$;
(12) ***end if***
(13) ***end for***;
(14) $T_{sav} = MG\text{-}HAG(S, G)$;
(15) ***return*** T_{sav}.

4.4 Service Clustering Algorithm

Compared with the K-means algorithm, the K-means++ algorithm has been optimized with the initial center selection strategy to obtain higher-quality clustering [34]. In the previous research work, we verified through experiments that on the same data set of SFVs, the clustering quality of K-means++ is slightly better than BIRCH, GMM, DBSCAN, and other algorithms, and the computational time complexity is lower than the above clustering methods [26].

In this paper, the K-means++ algorithm is exploited to construct a Web service clustering method called SEHA-KW that integrates semantic enhancement and heterogeneous association. Algorithm 3 details the processing steps of SEHA-KW. Firstly, Algorithm 3 invokes Algorithm 1 to generate the set of service function vectors T_{sfv} for all the Web services. Algorithm 2 is also employed to produce the set of service association vectors T_{sav}. Then, for each Web service in the set S, the algorithm concatenates its SFV and SAV into the service representation vector. Finally, the K-means++ algorithm is adopted to perform service clustering.

5 EXPERIMENTS AND ANALYSIS

In this section, we conduct experiments to test the following questions:

1. Is the quality of SFVs generated by SE-SimCSE better than other popular models?

Algorithm 3 SEHA-KW

Input: the set of Web services S, the set of service compositions SP, the number of clusters k;
Output: the set of service clusters SC;
(1) $T_{sfv} = SE\text{-}SimCSE(S)$;
(2) $T_{sav} = MG\text{-}HAG(S, SP)$;
(3) ***for each*** $s \in S$
(4) $srv(s) = sfv(s) \mid\mid sav(s)$;
(5) $SRV = SRV \cup \{srv(s)\}$;
(6) ***end for***
(7) $SC = K\text{-}means++(SRV, k)$;
(8) ***return*** SC

2. Can service heterogeneous associations improve the quality of Web service clustering?
3. Does SEHA-KW outperform the state-of-the-art service clustering methods?
4. What is the optimal dimension of the service representation vectors in service clustering?

5.1 Dataset and Experimental Setup

We have crawled 20 439 Web services and 6 218 Mashup services in the ProgrammableWeb. Here, Mashup services are used as the scenarios for service compositions. After deleting some services whose function description was too short, repeated registration or the number of services in the category was too small, 19 240 Web services were finally retained, belonging to 132 categories. A total of 4 773 Mashup services were retained, which included a total of 1 018 composite Web services belonging to 116 categories. All the texts of service descriptions are processed by case conversion, word segmentation, stop words removal, and stemming.

Table 1 presents the overview of our dataset. The frequently used incremental dataset construction methods in service clustering are employed to build the dataset [4, 13, 18]. Taking the main tag of the Web service as its category, the 19 240 services are divided into three data sets.

Dataset	Category-Top	Number
DS1	Top-20	9 393
DS2	Top-50	14 876
DS3	Top-132	19 240

Table 1. Overview of the dataset

5.2 Experimental Results

5.2.1 Evaluation Metrics

The commonly used metrics DBI, SC, AMI, NMI and Purity in the clustering quality evaluation are exploited to test the quality of service clustering [35, 36, 37]. Among the above metrics, except for DBI, which is characterized by a smaller value indicating a better clustering quality, all other metrics exhibit an improved clustering quality that is positively correlated with an increase in their values.

5.2.2 Performance Comparison for the Generation Models of SFVs

The performance of the proposed SE-SimCSE and the current popular generation models of SFVs are compared in this section. The comparison models are LDA [38], GSDMM [39], RoBERTa [40] and SimCSE [29]. LDA and GSDMM are the topic models, while RoBERTa and SimCSE are the neural network models.

Different models were used to generate SFVs for Web services in the DS1 to DS3, and the K-means++ algorithm was used to implement clustering. The values of each cluster evaluation metric are listed in Table 2. We can see that the SFVs generated by the LDA model have the worst service clustering quality, and the SFVs generated by GSDMM have significantly better clustering quality than LDA. This is mainly because GSDMM is suitable for extracting the topic features of short texts. The performance of RoBERTa and GSDMM is similar, and the clustering quality of SFVs generated by SimCSE is higher than that of LDA, GSDMM, and RoBERTa.

Dataset	Model	DBI	SC	AMI	NMI	Purity
DS1	LDA	1.717	0.311	0.272	0.282	0.292
	GSDMM	0.975	0.466	0.449	0.467	0.428
	RoBERTa	1.021	0.392	0.473	0.485	0.489
	SimCSE	0.949	0.494	0.529	0.537	0.525
	SE-SimCSE	**0.879**	**0.544**	**0.577**	**0.580**	**0.558**
DS2	LDA	1.946	0.306	0.267	0.281	0.296
	GSDMM	0.993	0.435	0.426	0.458	0.419
	RoBERTa	1.174	0.383	0.447	0.460	0.471
	SimCSE	0.972	0.488	0.490	0.513	0.502
	SE-SimCSE	**0.890**	**0.529**	**0.530**	**0.538**	**0.509**
DS3	LDA	1.683	0.340	0.273	0.284	0.301
	GSDMM	0.964	0.472	0.451	0.480	0.441
	RoBERTa	0.997	0.410	0.479	0.498	0.479
	SimCSE	0.937	0.507	0.530	0.548	0.541
	SE-SimCSE	**0.892**	**0.552**	**0.584**	**0.607**	**0.592**

Table 2. Performance comparison of different SFV generation models

SE-SimCSE is the generation method of SFVs proposed in this paper. It introduces a semantic enhancement mechanism based on the SimCSE framework. We

compare the performance of SE-SimCSE with SimCSE by aggregating the clustering evaluation data on the three datasets. In the five metrics of DBI, SC, AMI, NMI, and Purity, SE-SimCSE is 7.4 %, 9.1 %, 9.2 %, 7.9 %, and 5.8 % higher than SimCSE, respectively. Therefore, the proposed SE-SimCSE outperforms the currently popular models in terms of the generation quality of SFVs.

Back translation (BT), random deletion (RD), and random synonym substitution (RSS) are three common ways of semantic enhancement. Drawing on the third way, we propose a synonym substitution way for verbs (VSS) based on the textual features of service descriptions. To verify that the proposed semantic enhancement is better than others, we evaluate the service clustering quality generated by the SE-SimCSE under different semantic enhancement methods.

Dataset	Method	DBI	SC	AMI	NMI	Purity
DS1	BT	0.928	0.529	0.545	0.562	0.537
	RD	0.953	0.491	0.536	0.544	0.523
	RSS	0.906	0.535	0.568	0.571	0.545
	VSS	**0.879**	**0.544**	**0.577**	**0.580**	**0.558**
DS2	BT	0.930	0.502	0.523	0.523	0.498
	RD	0.932	0.492	0.514	0.520	0.492
	RSS	0.916	0.518	0.521	0.526	0.501
	VSS	**0.890**	**0.529**	**0.530**	**0.538**	**0.509**
DS3	BT	0.924	0.533	0.556	0.563	0.552
	RD	0.929	0.516	0.512	0.549	0.543
	RSS	0.911	0.542	0.565	0.584	0.564
	VSS	**0.892**	**0.552**	**0.584**	**0.607**	**0.592**

Table 3. Performance comparison of different semantic enhancement methods

Table 3 shows the metric values of service clustering generated under different semantic enhancement method on the three datasets. We can see that in DS1 to DS3, VSS has achieved the highest score value in all metrics. This verifies that our proposed semantic increment method is more effective than the other three methods.

5.2.3 Performance Evaluation of Heterogeneous Association in Improving Service Clustering Quality

The clustering method that uses SE-SimCSE to generate SFVs and combines them with the K-means++ algorithm is called SE-KW. In the SE-KW, service clustering does not consider service association. By using SE-KW, we construct the following comparison methods:

1. SEC-KW, which only integrates the collaboration association into SE-KW;
2. SET-KW, which only integrates the tag association into SE-KW.

We compare the performance of SE-KW, SEC-KW, SET-KW, and SEHA-KW (which combines tag association and collaboration association in the way of the HAG

Dataset	Model	DBI	SC	AMI	NMI	Purity
DS1	SE-KW	0.879	0.544	0.577	0.580	0.558
	SEC-KW	0.846	0.551	0.582	0.593	0.569
	SET-KW	0.772	0.595	0.614	0.644	0.621
	SEHA-KW	**0.748**	**0.605**	**0.621**	**0.642**	**0.623**
DS2	SE-KW	0.890	0.529	0.530	0.538	0.509
	SEC-KW	0.861	0.542	0.541	0.549	0.522
	SET-KW	0.793	0.589	0.588	0.588	0.588
	SEHA-KW	**0.776**	**0.593**	**0.618**	**0.639**	**0.620**
DS3	SE-KW	0.892	0.552	0.584	0.607	0.592
	SEC-KW	0.865	0.566	0.592	0.614	0.605
	SET-KW	0.796	0.603	0.623	0.642	0.652
	SEHA-KW	**0.751**	**0.617**	**0.628**	**0.649**	**0.625**

Table 4. Performance evaluation of service association on improving service clustering quality

graph) to evaluate the impact of introducing different types of service association on the clustering quality.

It can be seen from Table 4 that on DS1 to DS3 datasets, SEC-KW reduces the DBI by 3.34 % on average compared with SE-KW, and increases the SC, AMI, NMI, and Purity by 2.09 %, 1.42 %, 1.8 %, and 2.23 %, respectively. The data indicates that introducing collaboration associations can improve the clustering quality. However, the Mashup services in the dataset contain a limited number of Web services, resulting in a relatively low density of introduced collaboration associations, so we observe a small improvement in service clustering quality.

Compared with SE-KW, SET-KW decreases by 8.2 % on average in DBI, and increases by 7.71 %, 6.41 %, 6.72 %, and 9.73 % in SC, AMI, NMI, and Purity, respectively. The improvement of service clustering quality in each index is higher than SEC-KW, mainly because there are many tag associations in Web services. Therefore, it can be seen that with the increase in service association density, the quality of service clustering is significantly improved.

Among all the methods, SEHA-KW has obtained the highest score in the quality evaluation of service clustering. Compared with SE-KW, SEHA-KW decreases by 14.5 % on average in DBI, and increases by 11.69 %, 10.41 %, 11.88 % and 12.6 % in SC, AMI, NMI and Purity, respectively. According to the above values, the heterogeneous association by fusing the tag association and collaboration association significantly improves the quality of service clustering. SEHA-KW enhances the quality of service clustering significantly more than introducing tag association or collaboration association alone, indicating that the proposed heterogeneous association significantly improves the quality of service clustering.

The advantages of using heterogeneous association over using single tag association or collaboration association is that heterogeneous association improves service clustering quality from both positive or negative categorical relevance. Web ser-

vices with tag associations are positive categorical relevance and tend to fall into one category. Web services with collaboration association are negative categorical relevance, so they are more inclined to be divided into different categories. If only a single association is considered, two services are either positive or negative categorical relevance. Therefore, single service association either reduces or increases the distance between Web services when clustering.

However, according to statistics, 56.9 % of the Web services involved in Mashup services have tag associations. This means that tag association and collaboration association of Web services often exist simultaneously. Therefore, it is unreasonable to consider single tag association or collaboration association to improve the clustering quality. We should quantitatively consider the influence of tag association and collaboration association on service clustering quality from both positive and negative categorical relevance. Heterogeneous association can help solve the above problems, so it improves the quality of clustering better than tag association or collaboration association.

5.2.4 Comparison of SEHA-KW Method with Other Clustering Methods

The following state-of-the-art service clustering methods are selected for comparison to verify the advancement of the proposed method.

1. LFW + K [41]: This paper proposes a service vectorization method based on length feature weight, which takes into account parameters such as the dimension of Web service documents, the maximum frequency of terms in documents, and the number of terms in other documents to assign term weights accordingly to extract functional representations for service clustering.
2. GWSC [42]: This paper constructs a structural relationship graph and attribute bipartite graph corresponding to the structural relationship between Web services and their own attribute information. Random walk algorithm is used to obtain the structure context information and attribute context information of Web service. The Skip-gram model is employed to train the joint context to generate the service representation vectors for the classification and prediction of Web services.
3. KW-TG-SC [26]: An improved GSDMM model is proposed to overcome the problem of low quality of service representation vectors generated by traditional topic models. The service collaboration graph and Node2vec were used to vectorize the collaboration nodes. Finally, K-means++ was used to realize the service clustering based on the fusion of functional similarity and collaboration similarity.
4. UCSI-SC [27]: This paper uses Doc2vec to learn the functional representation of service description. The association network is established according to the service structure relationship. Tag sharing and collaboration are used as mutually exclusive associations to realize association representation learning. Unified

features are obtained by training the service classification model of partially tagged data, and a spectral clustering algorithm is used for service clustering.

Dataset	Model	DBI	SC	AMI	NMI	Purity
DS1	LFW + K	0.939	0.529	0.564	0.580	0.569
	GWSC	0.846	0.551	0.582	0.593	0.569
	KW-TG-SC	0.772	0.595	0.614	0.644	0.621
	UCSI-SC	0.821	0.584	0.592	0.616	0.609
	SEHA-KW	**0.748**	**0.605**	**0.621**	**0.642**	**0.623**
DS2	LFW + K	0.962	0.502	0.56	0.582	0.565
	GWSC	0.832	0.524	0.582	0.604	0.592
	KW-TG-SC	0.812	0.581	0.565	0.610	0.598
	UCSI-SC	0.827	0.579	0.601	0.623	0.613
	SEHA-KW	**0.776**	**0.593**	**0.618**	**0.639**	**0.620**
DS3	LFW + K	0.927	0.547	0.568	0.583	0.571
	GWSC	0.808	0.591	0.590	0.621	0.598
	KW-TG-SC	0.790	0.612	0.604	0.639	0.613
	UCSI-SC	0.819	0.599	0.615	0.652	0.619
	SEHA-KW	**0.751**	**0.617**	**0.628**	**0.649**	**0.625**

Table 5. Performance evaluation of service association on improving service clustering quality

The data in Table 5 shows that SEHA-KW obtained the highest scores in the five metrics on the three data sets. Therefore, the service clustering quality is better than comparison methods. Compared with the other four methods, DBI is reduced by 7.8 %-19.6 %, and SC is improved by 3 %-15 %. This shows that our method can improve the compactness of services within clusters and increase the distance between clusters when clustering. Meanwhile, the improvement intervals of AMI, NMI and Purity are 3.3 %-10.3 %, 2.1 %-10.1 % and 1.4 %-9.6 %, respectively, indicating that the accuracy of service clustering is significantly higher than that of the other four methods. The matching degree between the clustering results and the real categories of services is significantly improved. Compared with the other four methods, the proposed method has an average reduction of 13.7 % in DBI and an average increase of 9 %, 6.8 %, 6.1 % and 5.5 % in SC, AMI, NMI and Purity, respectively. Therefore, it can be concluded that the proposed method effectively improves the quality of service clustering.

Among the above four methods, LFW + K has the lowest clustering quality, which is mainly due to the lack of consideration about the context information of the terms in the service descriptions, and the failure to mine the implicit semantic relationships between service associations and complementary Web services.

GWSC employs the structurally connected graph and attributes bipartite graph to model the structure context and attribute context for Web services. The authors concatenate the node vectors of the two graphs for service clustering. GWSC

can effectively capture the implicit semantic relationship between Web services and present a high-quality service clustering.

KW-TG-SC considers both functional semantics and service associations and achieves relatively high scores in all indicators. Although it improves the GSDMM model which is suitable for short text topic extraction, it only considers the collaboration association between Web services and ignores tag association, which leads to insufficient mining of service association. It affects the extraction quality of association vectors and limits the improvement of the service clustering effect.

UCSI-SC is most similar to the method presented in this paper. It extracts features of Web services from functional semantics and service association. However, the performance of the Doc2Vec model used in this method is lower than that of the SE-SimCSE model adopted in our method when extracting functional semantics. In addition, tag association and collaboration association are regarded as two mutually exclusive associations. They are not fused together to quantitatively measure the association intensity between two services, which affects the improvement of service clustering quality.

5.2.5 Optimal Dimension of the Service Representation Vector

The dimension of service representation vector has an important influence on the quality of service clustering. In service clustering, it is often necessary to reduce the dimensionality of vectors to capture key information. If the dimension is too small, it is easy to cause information loss, resulting in the deterioration of clustering quality. On the contrary, if the dimension is too large, the feature space will be sparse and the clustering quality will be reduced. This section we will select a value of 32, 64, 128, 256, 512 and 768 as the optimal vector dimension through experiments.

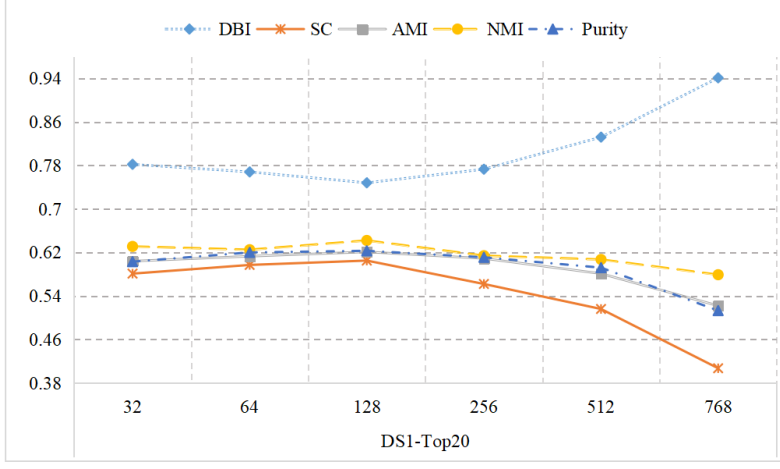


Figure 4. Metric values for different vector dimensions in DS1

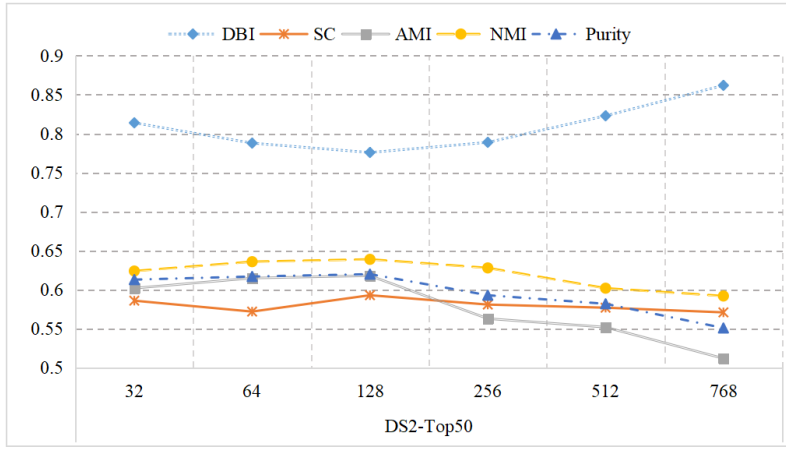


Figure 5. Metric values for different vector dimensions in DS2

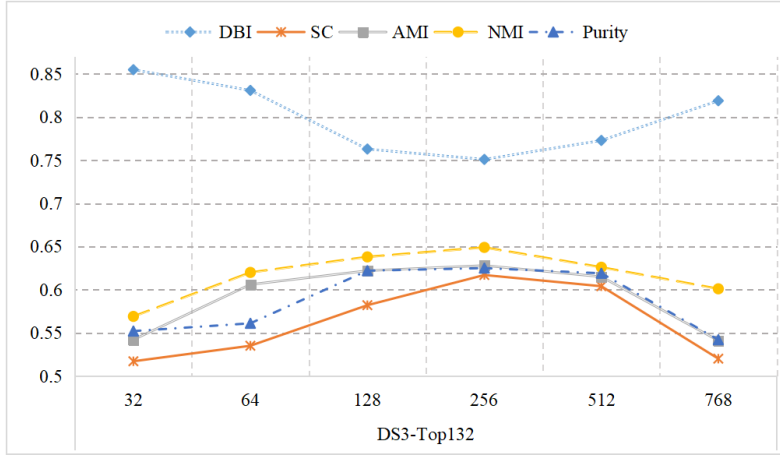


Figure 6. Metric values for different vector dimensions in DS3

Observing the curves in Figures 4, 5 and 6, in the three datasets, SC, AMI, NMI and Purity showed a trend of first increase and then decrease, while DBI showed a trend of first decrease and then increase. This means that there is a vector dimension for the best clustering quality in each dataset. It can be seen from the curve that when the number of service clusters is 20 and 50, the optimal dimension is 128. The clustering quality is the best when the number of clusters is 132 and the vector dimension is 256.

6 CONCLUSIONS

To improve the clustering quality of Web services, we propose semantic enhancement and heterogeneous correlation guided Web Service Clustering. With the help of SimCSE, a semantic-enhanced generation model for SFVs is constructed. A HAG fusing tag association and collaboration association is established. The walking strategy oriented to service association intensity is devised to sample node sequences in HAG. SAVs are generated by an improved GATNE model based on the sampling node sequences. Finally, service representation vectors, fused by SFVs and the association vectors, are used to perform service clustering by the K-means++ algorithm. Experiments show that the proposed service clustering method is significantly better than the state-of-the-art clustering methods in terms of clustering quality.

Future work mainly focuses on mining more types of service associations, such as the associations of service providers or geographic locations, to further expand the effect of service associations on improving the clustering quality.

REFERENCES

[1] Haorongbam, L.—Nagpal, R.—Sehgal, R.: Service Oriented Architecture (SOA): A Literature Review on the Maintainability, Approaches and Design Process. 2022 12th International Conference on Cloud Computing, Data Science and Engineering (Confluence), 2022, pp. 647–652, doi: 10.1109/Confluence52989.2022.9734153.

[2] Zhang, R.—Qi, L.—Yan, C.—Chen, Z.—Gong, W.—Xu, Y.—Rafique, W.: Privacy-Aware Web APIs Recommendation for Consumer Mashup Creation Based on Iterative Quantification. IEEE Transactions on Consumer Electronics, Vol. 71, 2025, No. 2, pp. 6460–6468, doi: 10.1109/TCE.2024.3372572.

[3] RapidAPI Website. RapidAPI, `https://rapidapi.com/hub`.

[4] Jalal, S.—Yadav, D. K.—Negi, C. S.: Web Service Discovery with Incorporation of Web Services Clustering. International Journal of Computers and Applications, Vol. 45, 2023, No. 1, pp. 51–62, doi: 10.1080/1206212X.2019.1698131.

[5] Jiang, R.—Xin, Y.—Chen, Z.—Zhang, Y.: A Medical Big Data Access Control Model Based on Fuzzy Trust Prediction and Regression Analysis. Applied Soft Computing, Vol. 117, 2022, Art. No. 108423, doi: 10.1016/j.asoc.2022.108423.

[6] Heidari, A.—Jafari Navimipour, N.: Service Discovery Mechanisms in Cloud Computing: A Comprehensive and Systematic Literature Review. Kybernetes, Vol. 51, 2022, No. 3, pp. 952–981, doi: 10.1108/K-12-2020-0909.

[7] Jiang, R.—Han, S.—Yu, Y.—Ding, W.: An Access Control Model for Medical Big Data Based on Clustering and Risk. Information Sciences, Vol. 621, 2023, pp. 691–707, doi: 10.1016/j.ins.2022.11.102.

[8] Fang, X.: Semantic Clustering Analysis for Web Service Discovery and Recognition in Internet of Things. Soft Computing, Vol. 27, 2023, No. 3, pp. 1751–1761, doi: 10.1007/s00500-021-06063-y.

[9] Agarwal, N.—Sikka, G.—Awasthi, L. K.: A Systematic Literature Review on Web Service Clustering Approaches to Enhance Service Discovery, Selection and Recommendation. Computer Science Review, Vol. 45, 2022, Art. No. 100498, doi: 10.1016/j.cosrev.2022.100498.

[10] Zhao, K.—Liu, J.—Xu, Z.—Liu, X.—Xue, L.—Xie, Z.—Zhou, Y.—Wang, X.: Graph4Web: A Relation-Aware Graph Attention Network for Web Service Classification. Journal of Systems and Software, Vol. 190, 2022, Art. No. 111324, doi: 10.1016/j.jss.2022.111324.

[11] Huang, Z.—Zhao, W.: A Semantic Matching Approach Addressing Multidimensional Representations for Web Service Discovery. Expert Systems with Applications, Vol. 210, 2022, Art. No. 118468, doi: 10.1016/j.eswa.2022.118468.

[12] Ghafouri, S. H.—Hashemi, S. M.—Hung, P. C. K.: A Survey on Web Service QoS Prediction Methods. IEEE Transactions on Services Computing, Vol. 15, 2022, No. 4, pp. 2439–2454, doi: 10.1109/TSC.2020.2980793.

[13] Shen, J.—Huang, W.—Hu, Q.: PICF-LDA: A Topic Enhanced LDA with Probability Incremental Correction Factor for Web API Service Clustering. Journal of Cloud Computing, Vol. 11, 2022, No. 1, Art. No. 19, doi: 10.1186/s13677-022-00291-9.

[14] Yang, D.—He, D.: Web Service Clustering Method Based on Word Vector and Biterm Topic Model. 2021 IEEE 6th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA), 2021, pp. 299–304, doi: 10.1109/ICCCBDA51879.2021.9442496.

[15] Cao, B.—Xiao, Q.—Zhang, X.—Liu, J.: An API Service Recommendation Method via Combining Self-Organization Map-Based Functionality Clustering and Deep Factorization Machine-Based Quality Prediction. Chinese Journal of Computers, Vol. 42, 2019, No. 6, pp. 1367–1383 (in Chinese).

[16] Agarwal, N.—Sikka, G.—Awasthi, L. K.: Evaluation of Web Service Clustering Using Dirichlet Multinomial Mixture Model Based Approach for Dimensionality Reduction in Service Representation. Information Processing and Management, Vol. 57, 2020, No. 4, Art. No. 102238, doi: 10.1016/j.ipm.2020.102238.

[17] Jiang, R.—Han, S.—Zhang, Y.—Chen, T.—Song, J.: Medical Big Data Access Control Model Based on UPHFPR and Evolutionary Game. Alexandria Engineering Journal, Vol. 6, 2022, No. 12, pp. 10659–10675, doi: 10.1016/j.aej.2022.03.075.

[18] Agarwal, N.—Sikka, G.—Awasthi, L. K.: WGSDMM+GA: A Genetic Algorithm-Based Service Clustering Methodology Assimilating Dirichlet Multinomial Mixture Model with Word Embedding. Future Generation Computer Systems, Vol. 145, 2023, pp. 254–266, doi: 10.1016/j.future.2023.03.028.

[19] Ye, H.—Cao, B.—Chen, J.—Liu, J.—Wen, Y.—Chen, J.: A Web Services Classification Method Based on GCN. 2019 IEEE Intl Conf on Parallel and Distributed Processing with Applications, Big Data and Cloud Computing, Sustainable Computing and Communications, Social Computing and Networking (ISPA/BDCloud/SocialCom/SustainCom), 2019, pp. 1107–1114, doi: 10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00158.

[20] Tang, B.—Yan, M.—Zhang, N.—Xu, L.—Zhang, X.—Ren, H.: Co-Attentive Representation Learning for Web Services Classification. Expert Systems with Ap-

plications, Vol. 180, 2021, Art. No. 115070, doi: 10.1016/j.eswa.2021.115070.

[21] Kang, G.—Xiao, Y.—Liu, J.—Cao, Y.—Cao, B.—Zhang, X.—Ding, L.: Tatt-BiLSTM: Web Service Classification with Topical Attention-Based BiLSTM. Concurrency and Computation: Practice and Experience, Vol. 33, 2021, No. 16, Art. No. e6287, doi: 10.1002/cpe.6287.

[22] Zhu, H.—Tan, W.—Yang, M.—Guo, K.—Li, J.: DSCPL: A Deep Cloud Manufacturing Service Clustering Method Using Pseudo-Labels. Journal of Industrial Information Integration, Vol. 31, 2023, Art. No. 100415, doi: 10.1016/j.jii.2022.100415.

[23] Zou, G.—Qin, Z.—He, Q.—Wang, P.—Zhang, B.—Gan, Y.: DeepWSC: Clustering Web Services via Integrating Service Composability Into Deep Semantic Features. IEEE Transactions on Services Computing, Vol. 15, 2022, No. 4, pp. 1940–1953, doi: 10.1109/TSC.2020.3026188.

[24] Ping, D. L.—Bing, G.—Wen, Z.: Web Service Clustering Approach Based on Network and Fused Document-Based and Tag-Based Topics Similarity. International Journal of Web Services Research (IJWSR), Vol. 18, 2021, No. 3, pp. 63–81, doi: 10.4018/IJWSR.2021070104.

[25] Cao, Y.—Liu, J.—Shi, M.—Cao, B.—Zhang, X.—Wang, Y.: Relationship Network Augmented Web Services Clustering. 2019 IEEE International Conference on Web Services (ICWS), 2019, pp. 247–254, doi: 10.1109/ICWS.2019.00050.

[26] Hu, Q.—Shen, J.—Wang, K.—Du, J.—Du, Y.: A Web Service Clustering Method Based on Topic Enhanced Gibbs Sampling Algorithm for the Dirichlet Multinomial Mixture Model and Service Collaboration Graph. Information Sciences, Vol. 586, 2022, pp. 239–260, doi: 10.1016/j.ins.2021.11.087.

[27] Kang, G.—Liu, J.—Xiao, Y.—Cao, Y.—Cao, B.—Shi, M.: Web Services Clustering via Exploring Unified Content and Structural Semantic Representation. IEEE Transactions on Network and Service Management, Vol. 19, 2022, No. 4, pp. 4082–4096, doi: 10.1109/TNSM.2022.3197725.

[28] Acheampong, F. A.—Nunoo-Mensah, H.—Chen, W.: Transformer Models for Text-Based Emotion Detection: A Review of BERT-Based Approaches. Artificial Intelligence Review, Vol. 54, 2021, No. 8, pp. 5789–5829, doi: 10.1007/s10462-021-09958-2.

[29] Gao, T.—Yao, X.—Chen, D.: SimCSE: Simple Contrastive Learning of Sentence Embeddings. In: Moens, M. F., Huang, X., Specia, L., Yih, S. W. (Eds.): Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021). Association for Computational Linguistics, 2021, pp. 6894–6910, doi: 10.18653/v1/2021.emnlp-main.552.

[30] Chen, Y.—Zou, X.—Zhang, J.—Yang, H.—Zhou, J.—Tang, J.: Representation Learning for Attributed Multiplex Heterogeneous Network. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '19), 2019, pp. 1358–1368, doi: 10.1145/3292500.3330964.

[31] Vaswani, A.—Shazeer, N.—Parmar, N.—Uszkoreit, J.—Jones, L.—Gomez, A. N.—Kaiser, Ł.—Polosukhin, I.: Attention Is All You Need. In: Guyon, I., Von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.): Advances in Neural Information Processing Systems 30 (NIPS

2017). Curran Associates, Inc., 2017, pp. 5998–6008, doi: 10.48550/arXiv.1706.03762.

[32] Xie, M.: Hierarchical Recommendation Algorithm Incorporated with Book Descriptions. 2021 3rd International Conference on Artificial Intelligence and Advanced Manufacture (AIAM), 2021, pp. 348–353, doi: 10.1109/AIAM54119.2021.00077.

[33] Pham, P.—Do, P.: W-Com2vec: A Topic-Driven Meta-Path-Based Intra-Community Embedding for Content-Based Heterogeneous Information Network. Intelligent Data Analysis, Vol. 24, 2020, No. 5, pp. 1207–1233, doi: 10.3233/IDA-194843.

[34] Li, H.—Wang, J.: Collaborative Annealing Power K-Means++ Clustering. Knowledge-Based Systems, Vol. 255, 2022, Art. No. 109593, doi: 10.1016/j.knosys.2022.109593.

[35] Mohanty, H.—Champati, S.—Barik, B. L. P.—Panda, A.: Cluster Quality Analysis Based on SVD, PCA-Based K-Means and NMF Techniques: An Online Survey Data. International Journal of Reasoning-Based Intelligent Systems, Vol. 15, 2023, No. 1, pp. 86–96, doi: 10.1504/IJRIS.2023.128368.

[36] Jha, P.—Tiwari, A.—Bharill, N.—Ratnaparkhe, M.—Mounika, M.—Nagendra, N.: A Novel Scalable Kernelized Fuzzy Clustering Algorithms Based on in-Memory Computation for Handling Big Data. IEEE Transactions on Emerging Topics in Computational Intelligence, Vol. 5, 2021, No. 6, pp. 908–919, doi: 10.1109/TETCI.2020.3016302.

[37] Choudhury, S. J.—Pal, N. R.: Deep and Structure-Preserving Autoencoders for Clustering Data with Missing Information. IEEE Transactions on Emerging Topics in Computational Intelligence, Vol. 5, 2021, No. 4, pp. 639–650, doi: 10.1109/TETCI.2019.2949264.

[38] Zhao, Y.—Qiao, Y.—He, K.: A Novel Tagging Augmented LDA Model for Clustering. International Journal of Web Services Research (IJWSR), Vol. 16, 2019, No. 3, pp. 59–77, doi: 10.4018/IJWSR.2019070104.

[39] Yin, J.—Wang, J.: A Dirichlet Multinomial Mixture Model-Based Approach for Short Text Clustering. Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14), 2014, pp. 233–242, doi: 10.1145/2623330.2623715.

[40] Briskilal, J.—Subalalitha, C. N.: An Ensemble Model for Classifying Idioms and Literal Texts Using BERT and RoBERTa. Information Processing and Management, Vol. 59, 2022, No. 1, Art. No. 102756, doi: 10.1016/j.ipm.2021.102756.

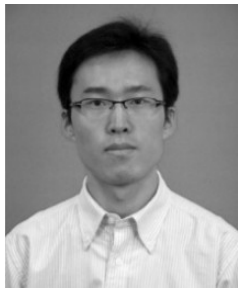
[41] Agarwal, N.—Sikka, G.—Awasthi, L. K.: Enhancing Web Service Clustering Using Length Feature Weight Method for Service Description Document Vector Space Representation. Expert Systems with Applications, Vol. 161, 2020, Art. No. 113682, doi: 10.1016/j.eswa.2020.113682.

[42] Xiao, Y.—Liu, J.—Hu, R.—Cao, B.—Cao, Y.: GAT2VEC-Based Classification Method for Web Services. Journal of Software, Vol. 32, 2021, No. 12, pp. 3751–3767, doi: 10.13328/j.cnki.jos.006102 (in Chinese).

Haoquan Qi is currently studying for a doctor's degree at the School of Computer Science and Technology, Donghua University, Shanghai, China. His research direction are services computing, serverless computing, and artificial intelligence. He has conducted research work on Mashup service clustering and recommendation, published papers and applied for patents.

Bing Wang is currently teaching at the Department of Information Engineering, Shandong Water Conservancy Vocational College, Rizhao, China. Her research direction are service computing and network security. She has published papers on service clustering and topic models.

Qiang Hu received his Ph.D. degree from the Shandong University of Science and Technology, Qingdao, China, in 2014. He is currently Associate Professor in the College of Information Science and Technology, Qingdao University of Science and Technology, Qingdao, China. He has taken in 8 projects supported by the National Nature Science Foundation, the National Key Basic Research Developing Program, and other projects at provincial levels. He has published over 20 papers in the international academic publications. His research interests are in formal method, service computing and text mining.



Pengwei Wang received his Ph.D. degree in computer science from the Tongji University, Shanghai, China, in 2013. He finished his postdoctoral research work at the Department of Computer Science, University of Pisa, Italy, in 2015. Currently, he is Associate Professor with the School of Computer Science and Technology, Donghua University, Shanghai, China. His research interests include cloud and edge computing, serverless, and services computing. He has published more than 90 papers on premier international journals and conferences, including IEEE Transactions on Services Computing, IEEE Transactions on Cloud Computing, IEEE Transactions on Systems, Man, and Cybernetics: Systems, IEEE Internet of Things Journal, etc.