IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 31, 2019, accepted July 1, 2019, date of publication July 3, 2019, date of current version July 19, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2926559

Semantic Sparse Service Discovery Using Word Embedding and Gaussian LDA

GANG TIAN[1], SHENGTAO ZHAO[1], JIAN WANG[2], ZIQI ZHAO[3], JUNJU LIU[4], AND LANTIAN GUO[5]

[1]School of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China
[2]State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China
[3]University of New South Wales, Sydney, NSW 2052, Australia
[4]Zhixing College, Hubei University, Wuhan 430011, China
[5]School of Automation and Electronic Engineering, Qingdao University of Science and Technology, Qingdao 266044, China

Corresponding author: Lantian Guo (guolt0211@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61702305 and Grant 61832014, in part by the China Postdoctoral Science Foundation under Grant 2017M622234, in part by the Qingdao City Postdoctoral Researchers Applied Research Projects, in part by the University of Science and Technology Program of Shandong Province under Grant J16LN08, in part by the Philosophy and Social Sciences Planning Project of the Ministry of Education under Grant 18YJC710032, and in part by the Scientific Research Project of the Education Department of Hubei Province of China under Grant B2018403.

ABSTRACT Nowadays, a growing number of web services are offered in API marketplaces browsed by service developers or third-party registries. Under this situation, API marketplaces' users greatly rely on a search engine to find suitable web services. However, due to the fact that functional attributes of web services are usually described in short texts, the search engine-based discovery approach suffers from the semantic sparsity problem, which hinders the effect of service discovery. To address this issue, we propose a novel web service discovery approach using word embedding and Gaussian latent Dirichlet allocation (Gaussian LDA). Unlike most existing service discovery approaches, our approach first uses context information generated by word embedding to enrich the semantics of service descriptions and users' queries. Then, the enriched service description is loaded into the Gaussian LDA model to acquire service description representation. Finally, the services are ranked by considering the relevance between the extended user's query and service description representation. The experiments conducted on a real-world web service dataset and the results demonstrate that the proposed approach achieves superior effectiveness on web service discovery.

INDEX TERMS Semantic sparsity, service discovery, word embedding, Gaussian LDA, service description representation.

I. INTRODUCTION

Benefit from the development of Internet infrastructure and the advantages of service-oriented computing (SOC), an increasing number of enterprises tend to exploit or convert their business applications into distributed web or cloud services [1], [2]. Furthermore, with social networks and cloud computing have become more and more popular, many new applications which combine web services from different sources are emerged. With this trend, it is important that functionality/data of services can be accessed for services' users remotely.

Service descriptions have been a significance perspective in concerning easy access to the IT service. Service descriptions, which act as a kind of individual Application Programming Interface (API) descriptions, allow users to invoke services without knowing how they are implemented. Once the service descriptions are released, services can be discovered [3], [4]. Nowadays, there are mainly two approaches to provide web services—-SOAP-based and REST-based. For the SOAP-based service, the Universal Description, Discovery and Integration (UDDI) standard was first proposed to address SOAP-based service discovery issue [3], [4]. Although UDDI has not been widely used, the significance of service discovery has attracted a great amount of attention. REST services discovery considers the provision of services destined to be consumed by other services (e.g. a machine-client). Similar to web APIs, REST services are generally accompanied with informal description documents (e.g. HTML pages), written in natural

The associate editor coordinating the review of this manuscript and approving it for publication was Zhanyu Ma.

language [3], [4]. As the natural language are widely used in descriptions, service discovery for natural language based descriptions has been an eager problem in service research field.

Service discovery for natural language based descriptions deeply relies on the search engine, which is mainly concentrated on traditional information retrieval (IR) techniques such as keyword matching. However, it may be faced with some recall problems due to decentralized registration, scarce keywords in service description, usage of synonyms or changes in keyword [5], [6]. There are two general approaches to alleviate these problems. One approach is to perform diverse searches and return various types of candidates, most of which may be unrelated to the search objective. To address the above issues, some research works extended the search query to achieve better discovery performance, e.g. [3] and [4]. Although they developed some effective query expansion schemes, there are no great efforts on enhancing the representation of description. The other approach is to reinforce the search performance by clustering services into functionally similar groups according to crawled service descriptions. Because of the effect on reducing the search space of the discovery process, the latter method has also attracted much attention to academic [7]–[12].

Although enhancing the search performance by clustering services has obtained significant success, with the growing number of services in recent years, many new problems have appeared. A considerable problem is semantics sparsity, which is caused by the short description text of services. That is to say, there is no sufficient explanation for completed semantics of services and not enough statistical information (e.g. the co-occurrence of vocabulary), which hinders effective text feature representation and further challenges the retrieval application [13], [14].

To address this issue, a lot of work has been put forward on how to enrich the representation of short text semantic by transferring external information. For example, Jin *et al.* [13] proposed a transfer learning method for short text clustering using auxiliary long text. Hu *et al.* [15] proposed a short text clustering method built on world knowledge. Above works generally enrich the representation of service descriptions based on an implicit assumption that the auxiliary information and object text is are semantically related. However, it is not easy to seek out such auxiliary information in the real world's text, which results in that assumption is not always reasonable. In addition, a great many of service discovery methods, which are involved in traditional information retrieval (IR) models, e.g. LDA, LSA, etc, frequently employ vector space as the feature representation. These methods may suffer from dimensionality curse due to the sparse representation of short text [16], [17].

To address this issue, we propose a method that integrating word embedding and Gaussian LDA (GLDA, for short) model to improve the service discovery performance [18]. The word embedding technique can capture the lexical semantic features in the text. In the embedded vector space, words with similar semantic and syntactic attributes tend to be close to each other [19]. Therefore, this characteristic not only can effectively model the context information such as word co-occurrence pattern, which is used for enriching the semantics of service descriptions, but also is particularly suitable for solving the problem of using synonyms/variants of keywords in queries. GLDA is an advanced topic model that considers the input document as a collection of embedding representations, and considers learned topics as multivariate Gaussian distributions in the embedding space. Thus, with GLDA model, the enriched service description representations can be effectively modeled as topics representations, which are further used for service clustering. Inspired by this, we incorporate word embedding and GLDA to achieve service discovery. The main contributions in this paper are listed as follows:

- A pre-trained word embedding set is leveraged to enrich the semantics of service descriptions. Then, the obtained continuous embedding representations are loaded into GLDA to learn the representations of service description text, aiming at improving the representation quality.
- Word embedding is also used for enriching the semantics of query reflecting service discovery requirement, which further facilitate the web service discovery process.
- The extensive experiment illustrates that the proposed approach outperforms other baseline approaches in the term of *Precision*, *Recall* and *F-Measure*.

The rest of the paper is organized as follows. The statement of related works about web service discovery is presented in Section II. The detail of proposed model is introduced in Section III. Empirical experiments and corresponding results are discussed and analyzed in Section IV. Finally, the conclusion of the paper is concluded in Section V.

II. RELATED WORK

Service discovery is considered as an important role in web or cloud computing application. Recent years, a great number of achievements have been obtained in this research domain.

Web service description languages can generally be divided into two categories: semantic-based and non-semantic-based. For example, two typical and commonly used types of semantic-based description are Ontology Web Language for Services (OWL-S) and Web Service Modeling Ontology (WSMO), while there are three typical non-semantic description types—-Web Services Description Language (WSDL) for SOAP-based services, Web Application Description Language (WADL) and natural language for REST-based services. Semantic-based approaches usually make the matching on high level [20]–[23], while non-semantic-based approaches usually employ the IR technique to index and retrieve relevant services [7], [24], [25]. In this paper, the goal of our approach is to develop for discovering web service with non-semantic description.

Since web services are usually described by different types, present non-semantics based discovery methods are various. Since a large number of traditional SOAP-based services

are described by WSDL documents, many non-semantics service discovery approaches focused on extracting useful content from WSDL documents as the input characteristic of the IR model. For instance, Liu and Wong [26] collect four functional elements (include service content, service context, host name and service name) from WSDL documents utilizing text mining technique, and cluster web services based on the four elements. In a similar way, Elgazzar *et al.* [7] also extract some function elements, including content, type, message, port and service name from WSDL documents as the input characteristic of the IR model for clustering web services. One issue of above works is that if web services are presented by other types of description language such as natural language, these WSDL-based approaches may not be effective or even work. In this paper, we concern about discovering the web service that the description of service is presented by natural language, since the REST-based services are increasingly used and usually presented by natural language.

Consequently, many research works have focused on improving service discovery ability in this area. Initial research works adapted traditional IR techniques, such as TF-IDF, VSM, and probabilistic models to achieve service discovery. For instance, Sajjanhar *et al.* [27] combined a typical TF-IDF algorithm with Singular Value Decomposition (SVD) to retrieve relevant services. SVD can reduce the dimensionality of the TF-IDF matrix, so that it can filter irrelevant services. Similarly, Elshater *et al.* [28] proposed a KDTree structure which combined a TF-IDF algorithm with VSM. Each node in the KDTree splits on a hyperplane dimension given by each term. Probabilistic models, e.g. Probabilistic Latent Semantic Analysis (PLSA), LDA, and extensions of these models, are considered to be effective ways to enhance the performance of service discovery [9], [29]. For instance, Chen *et al.* [9] proposed an augment LDA model by integrating not only WSDL but also tags information to boost the performance of web service clustering. It has been proved that incorporating external information can further reinforce the discovery performance.

However, due to the semantic sparsity of service descriptions, the present methods may not work effectively. To alleviate the semantic sparsity problem, a great number of works in the IR research domain have been proposed. Within these studies, integrating external information has been widely used as a good effect on improving the service discovery's performance. For instance, Hu *et al.* [15] employed world knowledge to obtain improved short text clustering result. Jin *et al.* [13] developed a transfer learning based approach for short text clustering by transferring the knowledge from auxiliary long texts. These methods can alleviate the semantic sparsity problem partially, however, there are also have some limitations. For instance, the work in [15] is based on an implicit assumption that the auxiliary data and the short texts are semantically related, which may be unreasonable in practice. In a similar way, the work [13] considers an implicit assumption that the topical structures of two domains are completely identical, however, that assumption would be unreasonable in practice. To tackle these problems, we integrate external context information learned by word embedding technique, which can boost the performance of some IR tasks [17], such as short text similarity measurement [16], [30].

In service discovery approaches based on probabilistic models, such as some LDA-based service discovery approaches, the basic assumption of these probabilistic models is that words are discrete polynomial distribution, and these models cannot benefit from the continuous word embedding vector. LDA [31] is a generative topic model designed to represent the hidden structure of a collection of documents. In LDA, it is assumed that every document in the corpus has a topic distribution, in which the discrete topic distribution is extracted from the symmetrical Dirichlet distribution.

Compared with LDA model, Gaussian LDA (GLDA, for short) model is proposed in [32] to model a set of words in a document as a sequence of embedded words rather than a sequence of word types. In GLDA [32], the input words are converted to continuous vectors, but not discrete values. Thus, each generated topic is presented as a multivariate Gaussian distribution. By analyzing the semantic similarity among embedded word vectors, it is proved that the parameterization of Gaussian is reasonable [32], which can help to incorporate word embedding set to improve topic modeling performance effectively. Therefore, instead of LDA, we use GLDA [32] to take advantage of word embedding and probabilistic models.

To our best knowledge, there is still no published service discovery approach, which incorporates word embedding technique with GLDA to address the semantic sparsity problem within it.

III. THE PROPOSED APPROACH

In this section, the framework of the proposed approach is described in subsection III-A. Then, the service description modeling using GLDA is illustrated in subsection III-B. The query modeling that integrates the word embedding is presented in subsection III-C. Finally, the service ranking, which is the final step of service discovery, is presented in subsection III-D.

A. FRAMEWORK

As shown in Fig. 1, the process of the proposed web service discovery approach consists of three major steps: service modeling, query modeling and service ranking.

1) In service modeling step, service descriptions that distributed over the Internet are firstly crawled and preprocessed. These service descriptions are used as input of the Word2vec model[1] to train word embedding. Alternatively, pre-trained word embedding set, such as trained word embedding set using general large scale corpus (i.e. Wikipedia[2])

[1] https://code.google.com/archive/p/word2vec/

[2] https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2

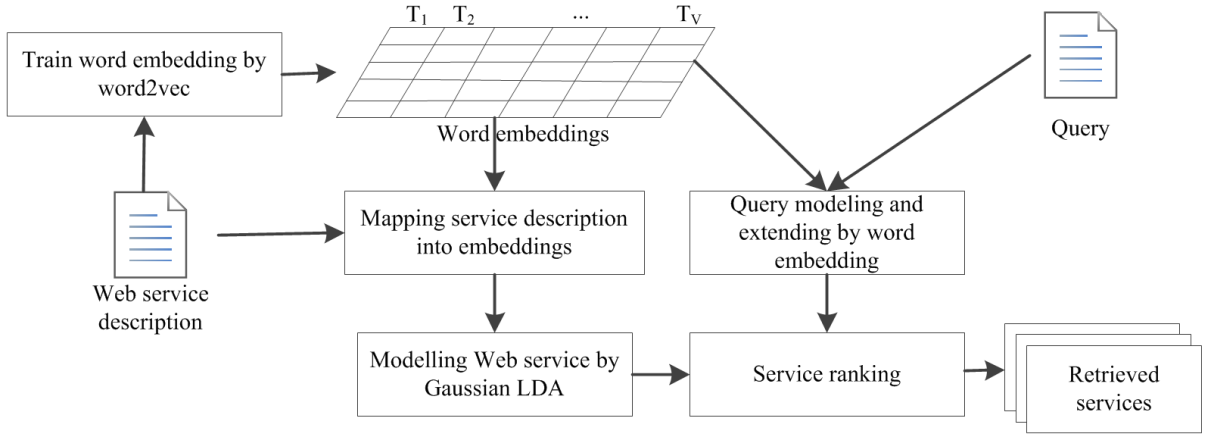


FIGURE 1. The framework of service discovery.

can also be introduced directly. After obtaining the word embedding set, the words in the service description can be mapped to embedding representation, which is further to serve as the input of GLDA. The effect of GLDA is to model each description text as the topic representation by hierarchies of latent factors.

2) In the query modeling step, the embedded set generated in the service modeling step also can be used to map the words in a given query to the embedding representation. Then, a query expansion algorithm is proposed, which expands queries by word embedding set, so as to find the similar neighbors of each word in queries. This algorithm can integrate more contextual information into the query, so that the semantics of query is enriched. After these processes, the extended query is completed.

3) In the service ranking step, a probabilistic service ranking model is proposed to retrieve relevant services for the user. This service ranking model is based on the hierarchies model and the extended query of the user. The ranking process is performed according to the relevance between services and user's query.

Please note that the step of service modeling, including training word embedding set, is performed offline, whereas the steps of query modeling and service ranking are conducted online, which is the retrieving stage of the service discovery. Hence, this paper is concentrated more on the accuracy performance or precision of service discovery, not the efficiency.

B. SERVICE DESCRIPTION MODELING USING GLDA

The foundation of GLDA model has been briefly introduced in Section II. Using GLDA modeling, service documents can be represented as random mixes of potential topics, and their proportions are extracted from Dirichlet Prior, which is similar to LDA model. The graphical representation of GLDA is shown in Fig. 2. Based on the above concepts, the GLDA generation process of service documents can be summarized as follows:

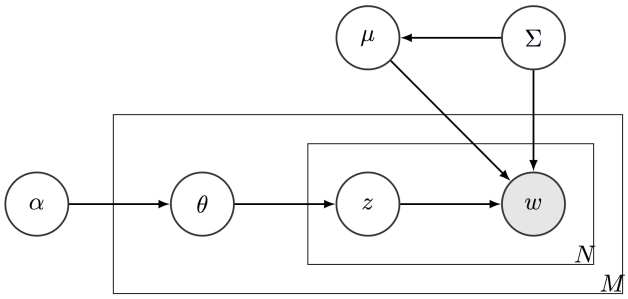


FIGURE 2. Graphical model of GLDA.

1) for topic $k = 1$ to K
   a) Draw topic covariance $\Sigma_k \sim W^{-1}(\Psi, \nu)$
   b) Draw topic mean $\mu_k \sim Normal(\mu, \frac{1}{k}\Sigma_k)$
2) for each service document d in corpus D,
   a) Draw topic distribution $\theta_d \sim Dir(\alpha)$
   b) for each word index i from 1 to N_d,
      i) Draw a topic $z_{(d,i)} \sim Multinomial(\theta_d)$
      ii) Draw a word $v_{(d,i)}$ with a probability: $v_{(d,i)}/z_{(d,i)}, u_{1..K}, \Sigma_{1..K} \sim Normal(\mu_{z_{(d,i)}}, \Sigma_{z_{(d,i)}})$

where k denotes each topic, d denotes each document, topic k is characterized as a multivariate Gaussian distribution with mean μ_k and covariance Σ_k, $Dir(\alpha)$ is the Dirichlet distribution.

The framework of GLDA based service description modeling is a hierarchically generative model, which is shown in Fig. 3. In this model, each word w in a web service description d can be represented by an embedded vector e, which is associated with the latent variable topic z. Each topic z is corresponding with the service description d. Based on the above two relationships, the service description generative model can be considered as two layers: the Service-Topic layer and the Topic-Embedding layer, corresponding two distributions: topic distribution and topic embedding distribution. As shown in Fig. 3, the hierarchical structure of GLDA is constructed using the above two distributions.

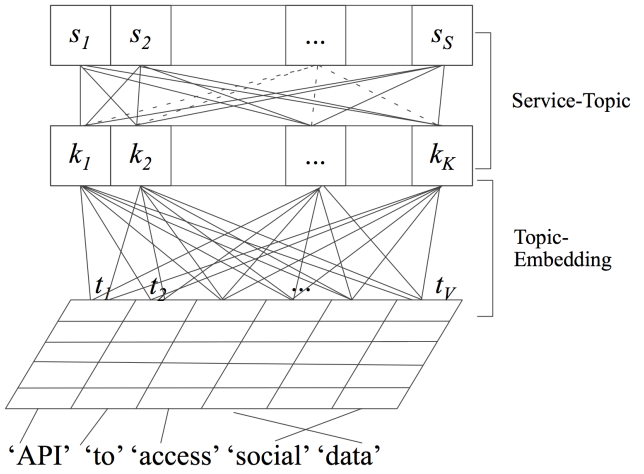


FIGURE 3. Hierarchical service description modeling using GLDA.

Specifically, each word w (e.g.,"API" and "access") is first represented as a fixed length vector after running Word2vec model. For instance, the word "academi" can be denoted as a vector that the size of it is 50—-[0.26 −0.30 −0.14 . . . −0.05]. Before using GLDA, each word in the service descriptions can be represented by a vector that trained by Word2vec. Therefore, the whole service description corpus can be mapped to a matrix fixed as 50 dimensions. Secondly, the matrix that represents the whole service description corpus is loaded into GLDA to generate two distributions: topic distribution and topic embedding distribution. Service topic distribution is originated from the parameter θ ($\theta \in |services| \times |topics|$) as a traditional LDA model.

In order to infer the posterior distribution of the topics over services and the topic assignments of individual words, we derive a collapsed Gibbs sampler to re-sample the above distribution or assignment iteratively, until the statuses of them are converged. The updating rules of iteration are shown in Equation 1.

$$p(z_{(d,i)} = k|z_{-(d,i)}, V_d, \zeta, \alpha) \propto (n_{(k,d)} + \alpha_k) \times t_{v_k - M+1}(v_{(d,i)}|\mu_k, \frac{\kappa_k + 1}{\kappa_k}\Sigma_k) \quad (1)$$

where $z_{-(d,i)}$ denotes the presently assigned topic for each word embedding in the word embedding set, excluding a service description which is at the i-th place of service description text set d. V_d denotes the embedding vector sequence for service description set d. $v_{(d,i)}$ denotes a vector sequence in a document d at position i. α denotes the parameters of the Dirichlet prior distribution. M denotes the length of individual word embedding vector. A tuple $\zeta = (\mu, \kappa, \Sigma, v)$ denotes the parameters of the prior distribution. In a M–dimensional space, each topic k is characterized as a multivariate Gaussian distribution with mean μ_k and covariance Σ_k. The multivariate t-distribution with freedom degree v, parameters μ, and parameters Σ can be expressed as $t_v(v|\mu, \Sigma)$. The parameters μ_k and Σ_k represent posterior mean and covariance, while κ_k and v_k represent the prior strength of posterior mean and covariance, respectively.

Note that the front unit of Equation 1 denotes the probability that the topic k would be assigned to the service description d. This unit is similar to LDA model, which means the process of generating topics from the service-topic distribution is similar. When running the GLDA, the first layer θ will be built using this unit.

The latter half of Equation 1 denotes the probability of topic k assigned to the word vector $v_{(d,i)}$. Given the current topic assignments, it can be regulated by a multivariate t-distribution with parameters $(\kappa_k, \mu_k, \Sigma_k, v_k)$, which is considered as a posterior distribution. The parameters of that posterior distribution are expressed as Equation 2.

$$\begin{aligned} \kappa_k &= \kappa + N_k \quad \mu_k = \frac{\kappa\mu + N_k\bar{v}_k}{\kappa_k} \\ v_k &= v + N_k \quad \Sigma_k = \frac{\Psi_k}{v_k - M + 1} \\ \Psi_k &= \Psi + C_k + \frac{\kappa N_k}{\kappa_k}(\bar{v}_k - \mu)(\bar{v}_k - \mu)^\top \end{aligned} \quad (2)$$

where the parameters $\bar{v}_k$ and C_k are calculated as:

$$\begin{aligned} \bar{v}_k &= \frac{\sum_d \sum_{i:z_{(d,i)}} v_{(d,i)}}{N_k} \\ C_k &= \sum_d \sum_{i:z_{(d,i)}=k} (v_{(d,i)} - \bar{v}_k)(v_{(d,i)} - \bar{v}_k)^\top \end{aligned} \quad (3)$$

Here $\bar{v}_k$ are the proportional forms allocated to topic k and C_k are average of sample covariance allocated to topic k, respectively. N_k is the total number of words allocated K for all description texts.

During Gibbs sampling process when the assignment probability of topic k to $v_{(d,i)}$ are computing, the updated parameters of the topic is needed to calculate as well. In sampling process, $\bar{v}_k$ can be updated from current value of $\bar{v}_k$. When a word embedding $v_{(d,i)}$ gets a new assignment to a topic k, then the new value of the topic covariance can be calculated by the current one, after updating κ_k, v_k and μ_k. After obtaining these parameters, discussed above topic embedding distribution can be simply conducted.

C. QUERY MODELING

Query modeling step is to map the user's submitted query into the feature space of word embedding. In query-based service retrieval process, since the heterogeneity of service authors/users, an effective way to alleviate these problems is to combine implicit semantics with explicit semantics [33]. Distributed word representation, mapping words to a dimensional feature continuous space, is considered as a type of semantic and syntactic representation of words [34]. In that continuous feature space, words with similar meanings have similar vectors. Thus, the continuous feature results in that synonym, near-synonym, semantic related, and context related words of an active word have a high probability to appear in its similar neighborhoods. For instance, in the view of word embedding trained by the Word2vec model, the first three words most similar to the word "month" are "minute",

"hour" and "day". According to this characteristic, we use the nearest neighbor in the embedded space to extend the query, since the query is usually short and semantic sparse. The query can be extended by appending the similar words, which is modeled by trained word embedding, to embody more context information.

The query extension process is shown in Algorithm 1. Specifically, a user's submitted query Q can be denoted by a set of words contained in the query: $Q = \{w_1, w_2, \ldots, w_{|Q|}\}$. In Algorithm 1, the three inputs are user's query Q, similarity threshold τ, and the embedding set E trained by Word2vec mode. The output of Algorithm is extended query Q_e. It consists of two units: the original query Q and the appended unit Ξ_w, which is the semantic enrichment of original query Q with the help of embedding set E. Ξ_w is represented as an intermediate vector for the final extended query vector Q_e. According to the word embedding characteristic, for each word in the query Q, its neighbor words whose similarity values are greater than prescribed threshold value τ would be added into the extended query Q_e.

Algorithm 1 Query Extension

Input: query Q, similarity threshold τ, the embedding set E
Output: the extended query Q_e.

$Q_e \leftarrow \varnothing$;
for *word* $w \in Q$ **do**
  for *word* $e_w \in E.top\_N\_similar(w)$ **do**
    $\Xi_w \leftarrow \varnothing$;
    if $similarity(e_w,w) \geq \tau$ **then**
      $\Xi_w \leftarrow \Xi_w \cup e_w$;
    end
    $Q_e \leftarrow Q_e \cup \Xi_w$;
  end
end
return Q_e

D. SERVICES RANKING

To rank the candidate web service according to a given query, we need to conduct a ranking algorithm to calculate the relevance between the user's query and the candidate web service, which indicates the matching degree between them. Inspired by the work in [35], we model the service ranking process as a probabilistic matching between query representation to the topic representation of service description.

The service ranking process relies on the generated probability to calculate the relevance. Specifically, generated probability of the service ranking process is defined as $P(Q|s_i)$, where Q denotes the set of words contained in the query, s_i denotes the i-th web service. Using the hypothesis of GLDA, $P(Q|s_i)$ can be calculated by Equation 4.

$$P(Q|s_i) = \prod_{e \in Q_e} P(e|s_i) = \prod_{e \in Q_e} \sum_{z=1}^{K} P(e|z)P(z|s_i) \qquad (4)$$

where the extended query of Q—-Q_e is gained from Algorithm 1. $P(e|z)$ and $P(z|s)$, which are the posterior probabilities, can be calculated according to Equation 2 and the matrix θ, respectively.

The most relevant service is the service that maximizes the conditional probability $P(Q|s_i)$ modeling the query. Therefore, related services are ranked according to their relevance scores with queries. Accordingly, we can get the ranking of queries by the retrieved services.

IV. EXPERIMENTS

In this section, we first introduce experimental setup including the experimental platform and dataset in subsection IV-A. Then, subsection IV-B introduces the evaluation metrics. In the experimental discussion, subsection IV-C illustrates our experimental design, along with observation of experimental results. In that subsection, we compared the proposed method with other baseline methods in terms of *Precision*, *Recall* and *F-Measure* metrics. Then, in subsection IV-D, to observe the influence of different parameters set on service discovery performance, we examined the results of conducting a series of the experiment with different parameter setting. Besides these, in order to observe the impact of our proposed query extension algorithm on service discovery performance, we conduct the experiment on validation of query extension in subsection IV-E. Finally, to observe the impact of different embedding set on service discovery performance, we conduct the experiment on validation of query extension in subsection in subsection IV-F.

A. EXPERIMENTAL SETUP

The experiment is conducted on a desktop with a 2.5 GHz Intel i5 Core 2 Duo CPU and a 16 GB RAM. We use a Python package named Gensim, running on *Ubuntu* operation system to train the word embeddings.[3] The program is developed by Python 2.7 and a Microsoft C++ hybrid environment. As for GLDA, we directly used the Java implementation in github.[4]

To evaluate the performance of the proposed approach, extensive experiments are conducted on a real-world web service discovery dataset called SAWSDL-TC3.[5] This dataset contains 1,043 service descriptions, and 42 queries that each has its corresponding relevance list as queried ground truth.

Since different corpus contains different context information in the word embedding sets and the scale of SAWSDL-TC3 dataset is small, in order to observe and compare the impact of pre-trained word embedding based on different corpus, we also adopt a general large scale corpus—-Wikipedia[6] to train the word embedding set. The statistics of the two corpora for training embedded sets are shown in Table 1.

[3]https://radimrehurek.com/gensim/models/word2vec.html
[4]https://github.com/rajarshd/Gaussian_LDA
[5]http://www.semwebcentral.org/projects/sawsdl-tc
[6]https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2

TABLE 1. Statistic of word embeddings.

	Wikipedia	SAWSDL-TC3
Number of Words	8,069,236	6,895
Number of Documents	3,758,076	1,043
Size of Embeddings	100	100

Before using SAWSDL-TC3 dataset, WSDL documents are pre-processed by following steps: Features extraction, Tokenization, Tag and stop words removal, Word stemming and Service Transaction Matrix construction (see [36] for more details). The word embedding training model for SAWSDL-TC3 and Wikipedia is based on Word2vec model in Python Gensim package. In the training implementation, the parameter setting for all the two corpora is: the size of the embedded word is 100, which denotes the vector dimension of the generated embedding representation, *window_size* is equal to 10, *min_count* is set to 5.

Note that the WSDL service dataset—-SAWSDL-TC3 is used to validate the proposed approach. There are two main reasons that the WSDL service description dataset is still used as a training corpus, rather than the short-text-based service description dataset. Firstly, as far as we can know, no standard test dataset for short-text-based web service discovery is presented, and SAWSDL-TC3 is a commonly used dataset in the web service discovery domain. Secondly, the semantic sparsity problem is solved by using embedded words to expand queries. A main purpose of the experiments is to validate the query extension effectiveness of the proposed approach. Since SAWSDL-TC3 contains the query and its corresponding relevant service list, it is able to evaluate the effectiveness of query extension.

B. EVALUATION METRICS

Precision, *Recall*, and *F-Measure* are employed as the evaluation metrics for the purpose of the evaluation. The three metrics are commonly used in information retrieval and ranking performance evaluation practices. Thus, this paper adopts them to evaluate service discovery performance. In SAWSDL-TC3 dataset, there are 42 different users' queries, in which a binary and hierarchical association set is provided for each query. The association set is considered as a set of service list that matches each query perfectly, which can be used to calculate *Precision*, *Recall*, and *F-Measure*. The larger *Precision*, *Recall*, and *F-Measure* are, the better the performances of the service discovery are. *Precision*, *Recall*, and *F-measure* are respectively defined as following:

- *Precision* represents the ration of the number of matched services in the Top-N ranking list to the length of the same list. It is shown as

$$Precision = \frac{\sum_{q \in OUT(Q)} |R(q) \cap T(q)|}{\sum_{q \in OUT(Q)} |T(q)|} \tag{5}$$

- *Recall* is the ratio of the number of matched services in the Top-N ranking list to the length of association set for a query in the dataset.

$$Recall = \frac{\sum_{q \in OUT(Q)} |R(q) \cap T(q)|}{\sum_{q \in OUT(Q)} |R(q)|} \tag{6}$$

- *F-Measure* is a harmonic mean of precision and recall which is shown as

$$F-Measure = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{7}$$

where N is the length of Top-N ranking list. Q is the set of different users' queries in dataset. $R(q)$ is association set in the dataset for q. $T(q)$ is the Top-N list of ranked services associated to a testing query q.

C. PERFORMANCE COMPARISON

To demonstrate the performance of our proposed method, we compare the proposed method with other baseline approaches. These approaches conducting service discovery are illustrated as following:

1) **TF-IDF**: In this approach, each service description is first represented by the TF-IDF algorithm, and then the relevance between query and TF-IDF based service description representation is calculated using Cosine similarity.
2) **PLSA** [29]: In this approach, PLSA is a technique used to model service description under probabilistic latent factors. The relevance between query and service description representation using learned latent factors is calculated using Cosine similarity.
3) **LDA** [29]: LDA is another commonly used latent factor based model. It is a Bayesian version of PLSA. LDA is to model the document using a three-layered structure, that is, the document contains a certain number of topics, and the words in each document are generated by the topic.
4) **Doc2vec** [37]: Doc2vec is an unsupervised algorithm to generate vectors for sentence or paragraph, which is essentially based on the principle of Word Embeddings. Doc2vec-based service clustering is to first vectorize documents, and then cluster the vector representation of service descriptions. Service descriptions in each cluster are treated as a topic. The relevance between query and service descriptions can be calculated by the distance of vector representation.
5) **WE-LDA** [38]: In this approach, the word vectors obtained by Word2vec are clustered into word clusters by Kmeans++ algorithm. Then, these word clusters as auxiliary information are incorporated to semi-supervise the LDA training process and learn the latent topic vectors of description documents. After obtaining the service description representation output from WE-LDA, we use it in the discovery process according to the method in subsection III-B except for extending the query.
6) **GLDA**: In this approach, we first learn embedding sets from the prepared corpus (e.g., SAWSDL-TC3,

Wikipedia) using word embedding model (Word2vec). Then, exchange the words in web service description files into embedding representation and take them as the input of the GLDA. After obtaining the service description representation output from GLDA, we use it in the discovery process according to the method in subsection III-B except for extending the query.

7) **GLDA + QE**: For GLDA + QE, we first take GLDA as the service description modeling method according to subsection III-B, and then adopt extending the query method according to the subsection III-C. Finally, incorporating service description modeling and extended query to generate service discovery result. The different between GLDA and GLDA + QE is that GLDA + QE uses the extended query in the discovery process.

Note that, for both PLSA and LDA models, we learn topics from the service description text according to the steps presented in the work [29], and adjust each parameter of algorithms to its optimal setting through cross-validation. In GLDA, the parameter α controls the weight contribution of the language model, while μ and Σ control the weight of input text data. As the work [32], the α, μ, and Σ parameters in this experiment are set as: $\alpha = 1/K$, $\mu =$ zero mean, and $\Sigma = 3 * I$, where I denotes the identity matrix. K denotes the number of topics in GLDA. Note that, for a fair comparison of selected methods, number of topics K in all of the methods is set to 6. Similarity Threshold τ, which controls the similar neighbors of an active word in extending the query is set to 0.96. Parameter tunings are present in subection IV-D. In addition, since Doc2vec is also based on word embedding technique, the vector size in Doc2vec approach is set as the same value with words representation dimensionality in GLDA. The impact of different parameter settings will be discussed in subsection IV-D.

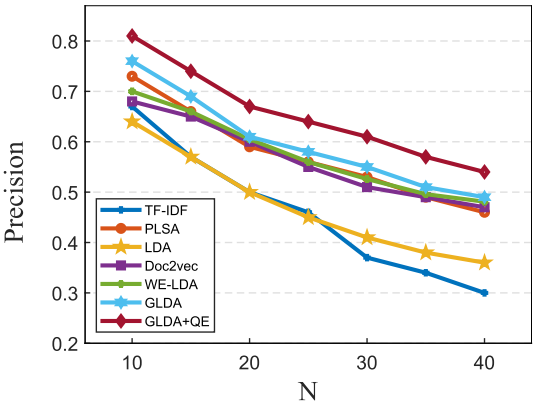


FIGURE 4. Comparison of performance on different methods in the term of *Precision*.

After extensive experiments, the obtained *Precision*, *Recall*, *F-Measure* results, and corresponding performance comparison of all methods are shown as Fig. 4, Fig. 5, Fig. 6 respectively. Apparently, with the number N of the retrieved

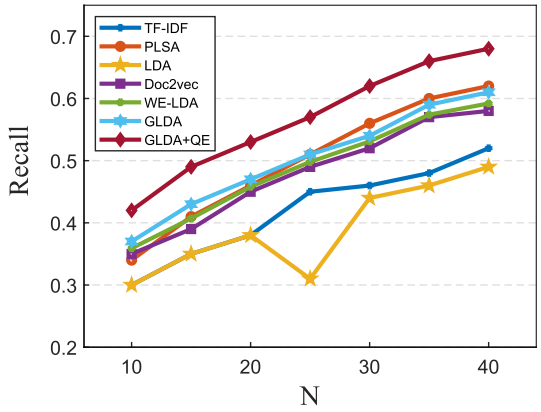


FIGURE 5. Comparison of performance on different methods in the term of *Recall*.

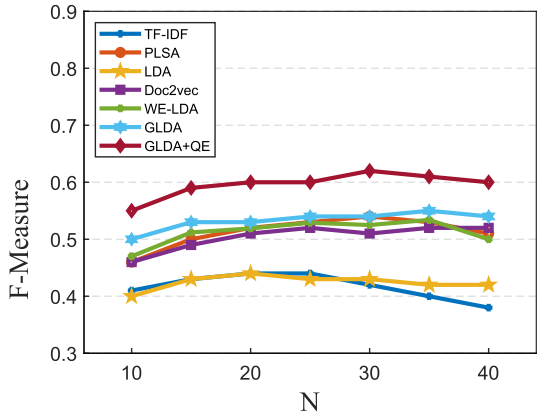


FIGURE 6. Comparison of performance on different methods in the term of *F-measure*.

service list increased gradually, the *Recall* and *F-Measure* metrics of all methods increase gradually, whereas the *Precision* metrics of them declines gradually. Since a larger value of N indicates more services would be matched with the ordered list in the dataset, the *Recall* metrics increases gradually. Since a larger value of N indicates more services would be unmatched with the ordered list in the dataset, the *Precision* metrics declines gradually. Further, two important observations are obtained as follows.

Observation 1: Firstly, as shown in Fig. 6, the *F-Measure* performance of the GLDA-based approach (GLDA and GLDA + QE) is superior to other approaches, which demonstrates the proposed approach is effective. Secondly, it also can be seen that the GLDA-based approach (without query expansion) has better performance than traditional latent factor based approach (LDA), word embedding based approach (Doc2vec) and their hybrid approach (WE-LDA). These results draw a conclusion that GLDA-based approach, which is on the basis of GLDA model and assisted by word embedding can capture more semantically consistent topics.

Observation 2: Another important observation is that using word embedding information to extend queries can help boost the service discovery performance. As shown in the Fig. 6, compared with the GLDA method, the GLDA + QE using

the Algorithm 1 to expand queries has better performance in all cases. One possible explanation is to introduce more query context information to enrich the semantics of queries.

D. IMPACT OF PARAMETERS

In this subsection, two parameters in our GLDA + QE model are discussed: Number of Topics K and Similarity Threshold τ. The parameter Number of Topics K influences topic modeling (GLDA) unit of our proposed model (GLDA + QE), while the parameter Number of Topics K influences query extension (QE) unit of our proposed model (GLDA + QE). Both of the two parameters can influence the proposed model, however, we cannot tune them collaboratively. The reason is that the two parameters influence the two units of the proposed model respectively.

Therefore, we tuned two parameters separately. We first tuned the parameter Number of Topics K. log-likelihood is the most common way to evaluate a probabilistic model, so we use log-likelihood to evaluate topic modeling performance of GLDA unit. Since log-likelihood only need to perform GLDA unit and does not need QE unit, the tuning two parameters separately can be achieved. In order to observe the impact of different Number of Topics K on topic modeling unit of our proposed model, we repeatedly executed experiments with different Number of Topics K to evaluate the log-likelihood result of GLDA model.

When the parameter Number of Topics K is fixed, the parameter Similarity Threshold τ can be tuned. In order to observe the impact of different Similarity Threshold τ on discovery performance, we repeatedly executed experiments with different Similarity Threshold τ based on the fixed Number of Topics K in the above experiment. When the discovery performance is optimal, the corresponding Similarity Threshold τ and fixed number of topic K can be seen as an optimal combination.

1) NUMBER OF TOPICS K

Here, relevant experiments are conducted to discuss the impact of the number of GLDA model topics. However, since no clear standard is provided for setting the number of topics, we repeatedly executed experiments with different number of topic K to evaluate the log-likelihood result of GLDA model. The experiment procedure follows the work in [39]. In order to obtain the optimized number of topics K, an estimation of $P(e|k)$ is computed with the different K value settings. For all values of k, running the GLDA model until the output converged. In that case, the log-likelihood values are finally stabilized within a few hundred of iterations.

Specifically, this experiment is performed with various settings of $K \in [3, 20]$, where the step length is 1, and the number of iterations is set to 300. Corresponding to each candidate values for the number of topics K, a group of the independent experiment is conducted. Fig. 7 shows the performance of log-likelihood. log-likelihood value increases as the topic number K increased from 1 to 6, whereas decreases and fluctuates when K changes from 6 to 20. When K is equal to 6, the better log-likelihood is obtained. Thus, in our experiments, the optimized number of topics is set to 6.

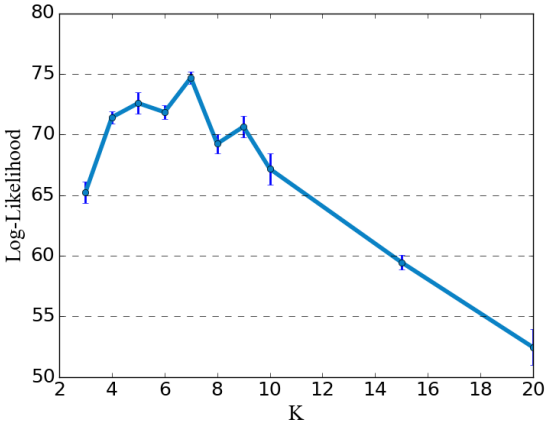


FIGURE 7. Impact of different number of topics on GLDA.

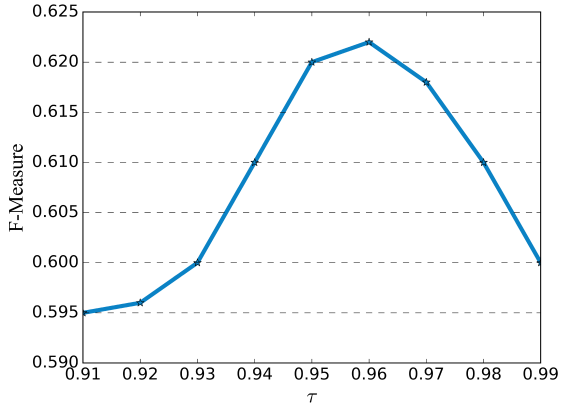


FIGURE 8. Impact of τ.

2) SIMILARITY THRESHOLD τ

As demonstrated in Algorithm 1, to control the similar neighbors of an active word in extending the query, a similarity threshold τ is employed to achieve that. An appropriate similarity threshold is crucial. A larger similarity threshold value not only means less similar words would be selected, but also may introduce more irrelevant contextual information into the extended queries. To find an appropriate similarity threshold, we tune it according to the performance of the proposed approach by cross-validation.

This experiment is performed with various settings of $\tau \in [0.9, 1]$, where the step length is 0.01. If the similarity threshold τ is very small, many very uncorrelated words would be appended into the extended query, which leads to the accuracy is very low. Thus, the similarity threshold τ is performed from 0.9 directly.

Fig. 8 shows the impact of different similarity threshold τ on discovery performance. As shown in Fig. 8, *F-Measure* value increases as the similarity threshold τ increased from 0.91 to 0.96, whereas decreases when the similarity threshold τ changes from 0.96 to 0.99. When τ is equal to 0.96,

an optimal *F-Measure* is achieved. Thus, in our experiments, similarity threshold τ is set to 0.96.

E. VALIDATION OF QUERY EXTENSION

Section III-C proposed a query expansion method to enrich the context information of the query. In order to observe the impact of our proposed query extension algorithm on service discovery performance, we conduct validation experiment of query extension in this subsection.

Specifically, we not only compare our GLDA + QE (GLDA based service discovery method with word embedding auxilliary Query Expansion) approach with GLDA (GLDA based service discovery method without query expansion) approach, but also compare our GLDA + QE approach with GLDA + WordNet (GLDA based service discovery method with WordNet auxilliary query expansion) and GLDA + NER (GLDA based service discovery method with NER auxilliary query expansion) approach. Similar to the schema of GLDA + QE, in GLDA + WordNet, WordNet[7] auxilliary query expansion was performed by adding synonyms, which is commonplace, leaving aside hyponyms or hypernyms. In GLDA + NER, NER (Named Entity Recognition)[8] auxilliary query expansion was performed by adding Named Entities combining constructed knowledge databases.

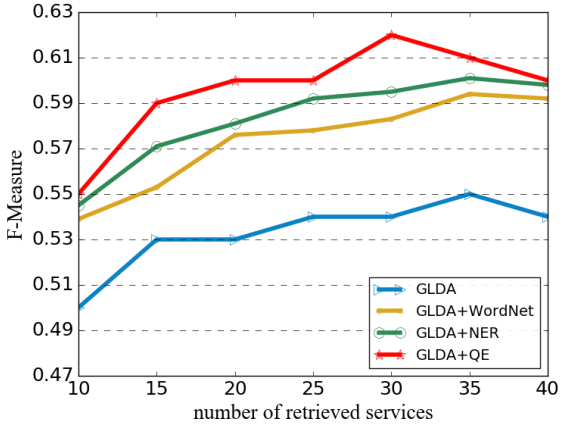


FIGURE 9. Impact of query extension on service discovery.

Fig. 9 shows *F-Measure* performance of these methods. It can be observed from the result that the *F-Measure* performance of GLDA + QE, GLDA + WordNet, and GLDA + NER are all superior to GLDA approach under all the setting of the number of retrieved services. The results demonstrate that integrating more background information to extend the query will help improve the service discovery performance. In addition, it also can be observed that GLDA + QE is better than GLDA + WordNet and GLDA + NER approach under all the setting of the number of retrieved services. The results demonstrate that service discovery method with word embedding auxilliary query expansion can obtain better performance than that with WordNet or NER auxilliary query expansion. However, it should be noted that our methods do not distinguish the relationship between additives and primitives, which will be further studied in our future work.

Moreover, in order to observe the ability of our proposed query extension approach on alleviating the semantics sparsity, we conduct the experiment to validate that in this subsection. Specifically, to simulate the different sparsity context, we truncated each WSDL file by randomly selecting words in accordance with a corresponding proportion. For instance, if we keep 10% words of a WSDL file to generate a new semantic sparsity WSDL file, the 90% words of that WSDL file is truncated.

To control the reserved proportion of each WSDL file, a parameter β is set. This experiment is performed with different settings of $\beta \in [10, 100]$, where the step length is 10. Note that, for a fair comparison, the number of retrieved services in all of the β settings is fixed to 30. Following the WSDL file truncation process discussed above, the similar retrieval task is repeatedly conducted to acquire the performance of the proposed approach in dealing with semantic sparsity problem.

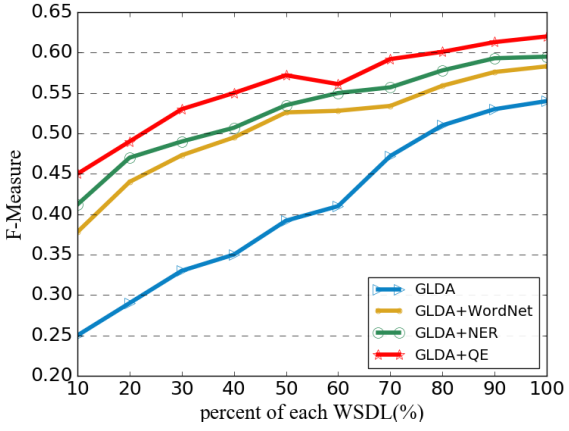


FIGURE 10. Validation of alleviating the semantics sparsity.

As shown in Fig. 10, it also shows that *F-Measure* performance of GLDA + QE, GLDA + WordNet, and GLDA + NER approach under all β settings is better than that of GLDA, which is without query expansion. Moreover, *F-Measure* performance of GLDA + QE approach under all β settings is better than that of GLDA + WordNet and GLDA + NER approach. All of these results indicate that the query expansion processing of the proposed model can boost the retrieve tasks effectively. More importantly, the greater the percentage of WSDL participation, the better the performance obtained. It is verified that the proposed model can deal with the problem of semantic sparsity well.

F. VALIDATION OF EMBEDDING SET

The embedding set, which is trained from embedding training corpus, plays two significant roles: one is to transform the Bag of Words (BOW) model into the continuous embedding

[7]http://www.nltk.org/howto/wordnet.html
[8]https://www.nltk.org/book/ch07.html

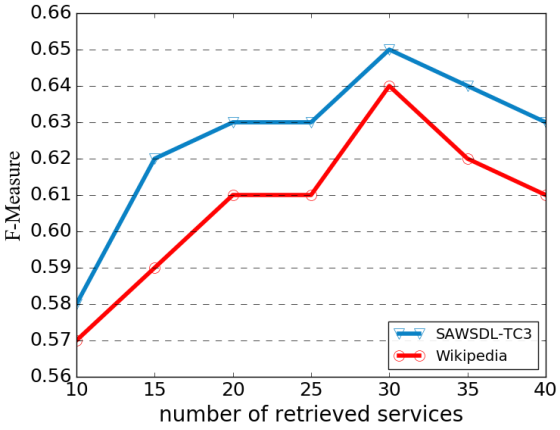


FIGURE 11. **Impact of different embedding sets.**

space, the other is to extend the query by appending additional context information. To observe the impact of different embedding set on service discovery performance, we conduct validation of embedding set experiment in this subsection. Fig. 11 shows the *F-Measure* performance of different word embedding set trained by Word2vec model using two different corpora—-SAWSDL-TC3 and Wikipedia respectively, as described in Section IV-A.

In view of the above experiment results, it can be obviously seen that the embedding set employing SAWSDL-TC3 corpus obtains better performance than the embedding set employing Wikipedia corpus, in spite of the performance promotion is not tremendous. There are two possible reasons:

1) SAWSDL-TC3 trained embedding set may be much more domain-specific than Wikipedia trained embedding set. That is to say, the words in web services description (SAWSDL-TC3) belong to the application domain, and they should have different word usage and distribution, compared with Wikipedia.

2) Some words, which are parsed from the WSDL files, do not have sufficient appearance time in Wikipedia corpus. Consequently, although these words are very informative, they are removed when training the embedding set. For instance, the term ''lendingduratio'' composed of words ''lending'' and ''duration'' is informative in lots of description texts, but it is overlooked in the embedded word representation since it is not parsed into individual words.

V. CONCLUSION

To address the semantic sparse problem in discovering service, this paper proposed a novel service discovery method combining GLDA (Gaussian LDA) and word embedding. Firstly, the word embedding technique is employed to generate the embedding set for all words in web service description text, since embedding based representation can embody effectively context features. Then, the semantics of service description is enriched by using words whose have similar semantic and syntactic attribute with the active word. After that, the enriched service description is loaded into GLDA model to train service description text representation. Moreover, word embedding technique is also used for enriching the semantics of the service discovery query, which further facilitates the web service discovery as well. The candidate services are ranked by considering the relevance between the extended user's query and service description text representation.

Experiments conducted on a real-world web service dataset, and the results demonstrate that the proposed approach achieves superior effectiveness on web service discovery, which means that our method is feasible, especially adding meaningful words in the discovery process.

In the future, we would like to further study the usefulness of various etymologies embedded in web service discovery. We also try to extract and utilize more description text features to achieve goal-oriented web service discovery. Besides these, we would like to employ more properties and information to achieve advanced models as well.

ACKNOWLEDGMENT

(Gang Tian and Lantian Guo are co-first authors.)

REFERENCES

[1] A. Zhou, S. Wang, B. Cheng, Z. Zheng, F. Yang, R. N. Cheng, M. R. Lyu, and R. Buyya, "Cloud service reliability enhancement via virtual machine placement optimization," *IEEE Trans. Serv. Comput.*, vol. 10, no. 6, pp. 902–913, Nov./Dec. 2017.

[2] S. Wang, Q. Zhao, N. Zhang, T. Lei, and F. Yang, "Virtual vehicle coordination for vehicles as ambient sensing platforms," *IEEE Access*, vol. 6, pp. 11940–11952, 2018.

[3] I. Lizarralde, C. Mateos, J. M. Rodriguez, and A. Zunino, "Exploiting named entity recognition for improving syntactic-based Web service discovery," *J. Inf. Sci.*, vol. 45, pp. 398–415, Mar. 2018.

[4] I. Lizarralde, J. M. Rodriguez, C. Mateos, and A. Zunino, "Word embeddings for improving REST services discoverability," in *Proc. 43rd Latin Amer. Comput. Conf. (CLEI)*, Sep. 2017, pp. 1–8.

[5] J. Wang, P. Gao, Y. Ma, K. He, and P. C. K. Hung, "A Web service discovery approach based on common topic groups extraction," *IEEE Access*, vol. 5, pp. 10193–10208, 2017.

[6] C. Wang, Z. Luo, X. Zhang, K. He, and X. Chen, "An approach to business process registration for enterprise collaboration: Using BPEL as an example," *Int. J. Bus. Process Integr. Manage.*, vol. 7, no. 3, pp. 181–196, 2015.

[7] K. Elgazzar, A. E. Hassan, and P. Martin, "Clustering WSDL documents to bootstrap the discovery of Web services," in *Proc. 17th IEEE Int. Conf. Web Services*, Jul. 2010, pp. 147–154.

[8] L. Chen, L. Hu, Z. Zheng, J. Wu, J. Yin, Y. Li, and S. Deng, "WTCluster: Utilizing tags for Web services clustering," in *Service-Oriented Computing*. New York, NY, USA: Springer, 2011, pp. 204–218.

[9] L. Chen, Y. Wang, Q. Yu, Z. Zheng, and J. Wu, "WT-LDA: User tagging augmented LDA for Web service clustering," in *Service-Oriented Computing*. New York, NY, USA: Springer, 2013, pp. 162–176.

[10] Y. Zhao, K. He, and Y. Qiao, "ST-LDA: High quality similar words augmented LDA for service clustering," in *Proc. 18th Int. Conf. Algorithms Archit. Parallel Process.*, Dec. 2018, pp. 46–59.

[11] L. D. J. Silva, D. B. Claro, and D. C. P. Lopes, "Semantic-based clustering of Web services," *J. Web Eng.*, vol. 14, nos. 3–4, pp. 325–345, Jul. 2015.

[12] A.-P. Zhao, L. Yu, and W.-M. Yang, "Semantically structured service community discovery: Based on relationship and functionality," *Int. J. Comput. Sci. Eng.*, vol. 13, no. 3, pp. 233–245, 2016.

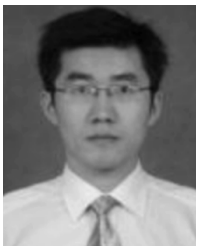
[13] O. Jin, N. N. Liu, K. Zhao, Y. Yu, and Q. Yang, "Transferring topical knowledge from auxiliary long texts for short text clustering," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2011, pp. 775–784.

[14] S. Seifzadeh, A. K. Farahat, M. S. Kamel, and F. Karray, "Short-text clustering using statistical semantics," in *Proc. 24th Int. Conf. World Wide Web Companion*, May 2015, pp. 805–810.

[15] X. Hu, N. Sun, C. Zhang, and T.-S. Chua, "Exploiting internal and external semantics for the clustering of short texts using world knowledge," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, Nov. 2009, pp. 919–928.
[16] T. Kenter and M. de Rijke, "Short text similarity with word embeddings," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2015, pp. 1411–1420.
[17] J. Xie, Z. Song, Y. Li, Y. Zhang, H. Yu, J. Zhan, Z. Ma, Y. Qiao, J. Zhang, and J. Guo, "A survey on machine learning-based mobile big data analysis: Challenges and applications," *Wireless Commun. Mobile Comput.*, vol. 2018, Aug. 2018, Art. no. 8738613.
[18] N. Zhang, J. Wang, Y. Ma, K. He, Z. Li, and X. Liu, "Web service discovery based on goal-oriented query expansion," *J. Syst. Softw.*, vol. 142, no. 8, pp. 73–91, Aug. 2018.
[19] T. Mikolov, W.-T. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proc. HLT-NAACL*, 2013, pp. 746–751.
[20] M. Klusch, B. Fries, and K. Sycara, "Automated semantic Web service discovery with OWLS-MX," in *Proc. 5th Int. Joint Conf. Auto. Agents Multiagent Syst.*, May 2006, pp. 915–922.
[21] N. Zhang, J. Wang, K. He, and Z. Li, "An approach of service discovery based on service goal clustering," in *Proc. IEEE Int. Conf. Services Comput.*, Jul. 2016, pp. 114–121.
[22] P. Rodriguez-Mier, C. Pedrinaci, M. Lama, and M. Mucientes, "An integrated semantic Web service discovery and composition framework," *IEEE Trans. Services Comput.*, vol. 9, no. 4, pp. 537–550, Aug. 2015.
[23] S. Pan, J. Wu, X. Zhu, G. Long, and C. Zhang, "Boosting for graph classification with universum," *Knowl. Inf. Syst.*, vol. 50, no. 1, pp. 53–77, Jan. 2017.
[24] R. Nayak and B. Lee, "Web service discovery with additional semantics and clustering," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell.*, Nov. 2007, pp. 555–558.
[25] J. Wu, L. Chen, Z. Zheng, M. R. Lyu, and Z. Wu, "Clustering Web services to facilitate service discovery," *Knowl. Inf. Syst.*, vol. 38, no. 1, pp. 207–229, 2014.
[26] W. Liu and W. Wong, "Web service clustering using text mining techniques," *Int. J. Agent-Oriented Softw. Eng.*, vol. 3, no. 1, pp. 6–26, Feb. 2009.
[27] A. Sajjanhar, J. Hou, and Y. Zhang, "Algorithm for Web services matching," in *Proc. 6th Asia–Pacific Web Conf.*, May 2004, pp. 665–670.
[28] Y. Elshater, K. Elgazzar, and P. Martin, "goDiscovery: Web service discovery made efficient," in *Proc. 22th IEEE Int. Conf. Web Services*, Jul. 2015, pp. 711–716.
[29] G. Cassar, P. Barnaghi, and K. Moessner, "Probabilistic methods for service clustering," in *Proc. 4th Int. Workshop Semantic Web Service*, Nov. 2010, pp. 1–17.
[30] L. Yu, Z. Junxing, and S. Y. Philip, "Service recommendation based on topics and trend prediction," in *Proc. 12th EAI Int. Conf. Collaborative Comput. Netw., Appl. Worksharing*, 2016, pp. 343–352.
[31] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.
[32] R. Das, M. Zaheer, and C. Dyer, "Gaussian lda for topic models with word embeddings," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2015, pp. 795–804.
[33] J. Tekli, "An overview on XML semantic disambiguation from unstructured text to semi-structured data: Background, applications, and ongoing challenges," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 6, pp. 1383–1407, Jun. 2016.
[34] Y. Li, L. Xu, F. Tian, L. Jiang, X. Zhong, and E. Chen, "Word embedding revisited: A new representation learning and explicit matrix factorization perspective," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 3650–3656.
[35] M. Aznag, M. Quafafou, and Z. Jarir, "Leveraging formal concept analysis with topic correlation for service clustering and discovery," in *Proc. 21st IEEE Int. Conf. Web Services*, Jul. 2014, pp. 153–160.
[36] M. Aznag, M. Quafafou, and Z. Jarir, "Correlated topic model for Web services ranking," *Int. J. Adv. Comput. Sci. Appl.*, vol. 4, p. 6, Jun. 2013.
[37] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. 31st Int. Conf. Mach. Learn.*, Jun. 2014, pp. 1188–1196.
[38] M. Shi, J. Liu, D. Zhou, M. Tang, and B. Cao, "WE-LDA: A word embeddings augmented LDA model for Web services clustering," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jun. 2017, pp. 9–16.
[39] W. Buntine, "Estimating likelihoods for topic models," in *Proc. 6th Asian Conf. Mach. Learn.* New York, NY, USA: Springer, 2009, pp. 51–64.

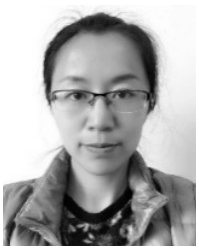
GANG TIAN received the Ph.D. degree in computer science and engineering from Wuhan University, China, in 2016. He is currently an Assistant Professor with the School of Computer Science and Engineering, Shandong University of Science and Technology, China. His current research interests include web service discovery, transfer learning, feature engineering, and knowledge extraction.

SHENGTAO ZHAO is currently a graduate student with the School of Computer Science and Engineering, Shandong University of Science and Technology, China. His research interests include web service discovery and feature engineering.



JIAN WANG received the Ph.D. degree from Wuhan University, China, in 2008, where he is currently a Lecturer with the State Key Laboratory of Software Engineering. His current research interests include software engineering and services computing.

ZIQI ZHAO is currently a graduate student with the University of New South Wales, Australia. Her research interests include web service discovery and feature engineering.



JUNJU LIU is currently an Associate Professor with the Zhixing College, Hubei University, Wuhan, China. Her current research interests include services computing and mathematical modeling.

LANTIAN GUO is currently pursuing the Ph.D. degree with the School of Automation, Northwestern Polytechnical University, China. He was a Visiting Research Student with the Qingdao University of Science and Technology, and with the School of Computing, Queen's University, Canada. His current research interests include big data, recommendation systems, machine learning, and artificial intelligence.