International Journal of Web Services Research
Volume 16 • Issue 1 • January-March 2019

Augmenting Labeled Probabilistic Topic Model for Web Service Classification

Shengye Pang, School of Computer Engineering and Science, Shanghai University, Shanghai, China

Guobing Zou, School of Computer Engineering and Science, Shanghai University, Shanghai, China

Yanglan Gan, School of Computer Science and Technology, Donghua University, Shanghai, China

Sen Niu, School of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai, China

Bofeng Zhang, School of Computer Engineering and Science, Shanghai University, Shanghai, China

ABSTRACT

Web service classification has become an urgent demand on service-oriented applications. Most existing classification algorithms mainly rely on the original service descriptions. That leads to low classification accuracy, since it cannot fully reflect the semantic feature specific to a service category. To solve the issue, this article proposes a novel approach for web service classification, including service topic feature extraction, service functionality augmentation, and service classification model learning. The characteristic is that the original service descriptions can be semantically augmented, which is fed to deriving a service classifier via labeled probabilistic topic model. A benefit from this approach is that it can be applied to an online service management platform, where it assists service providers to facilitate the registration process. Extensive experiments have been conducted on a large-scale real-world data set crawled from ProgrammableWeb. The results demonstrate that it outperforms state-of-the-art methods in terms of service classification accuracy and convergence speed.

KEYWORDS

Labeled LDA, *Service Classification*, *Service Feature Extraction*, *Service Functionality Augmentation*, *Web Service*

1. INTRODUCTION

Due to the fast advancement of Web 2.0 technologies and service-oriented computing, more and more service providers publish their services on the internet mainly in the form of web APIs. They can be more easily organized and manipulated in a loosely coupled style for creating service mashups to fulfill comprehensive functional requirements and offer value-added integrated software systems with complex business processes. As the rapid increase in the number and diversity of web services, it accelerates the interoperable machine-to-machine interaction and greatly promotes the procedure of service discovery, optimum selection, automatic composition and recommendation (Xia, Luo, Li, & Zhu, 2013; Xia, Liu, Liu, & Zhu, 2012; Li, Luo, Xia, Han, & Zhu, 2015). However, with the boom of overwhelming number of functional characteristics of the published web services, there are

DOI: 10.4018/IJWSR.2019010105

always hundreds of categories in an online RESTful service repository. As a result, it tends to be a labor-intensive challenging task for service providers to search and find an appropriate category from diverse registered ones, when publishing their API services on a service management platform. For example, ProgrammableWeb.com, which is the largest online RESTful service repository (APIs and mashups), collects over 19,000 APIs and 7000 mashups with more than 400 diverse categories on the web. In addition to providing basic registration information when service providers register their API services on ProgrammableWeb, it needs to further manually choose at least one desired category from more than 400 categories so that it can match corresponding service functional description. Therefore, how to design an effective approach that can classify web services and recommend an accurate category has become a critical research issue to be addressed (Ames & Naaman, 2007).

In recent years, correlative research efforts have been posed on web service classification (Tsoumakas, Katakis, & Taniar, 2008). These existing approaches achieve the goal of web service classification and service tag recommendation by training traditional supervised learning model (e.g., SVM) (Lopez & Maldonado, 2016; Wang, Shy, Zhou, & Bouguettaya, 2010), active learning-based supervised learning model (Tong & Koller, 2001; Liu, Agarwal, Ding, & Yu, 2016; Shi, Liu, & Yu, 2017), or a comprehensive supervised learning model where unlabeled probabilistic topic model (e.g., LDA) (Krestel, Fankauser, & Nejdl, 2009) has been applied to extract semantic feature of web services. Some of the works generally learn a classification model under an existing labeled service repository, while active learning method was taken into account for boosting the learned service classifier, where the most informative services are intellectually selected at each iteration and manually labeled with human efforts to enrich the quality of small scale training data. Although they take advantage of the existing service repository as training data to derive a service classifier which can be easily deployed and applied, it is still unsatisfactory for service providers' demands on high accuracy of web service classification.

The essential reason is that existing approaches have deficiencies on their effectiveness and efficiency. More specially, the disadvantages of current paradigm for web service classification are twofold. (1) On one hand, they mainly rely on the original service descriptions for learning a service classifier, where each functional description of a RESTful web service only consists of a bunch of short text (e.g., 10 to 20 words), failing to be fully understood on its corresponding category. Furthermore, it is observed that some words are frequently repeated with high occurrence across different service descriptions, which obviously disturbs the purity of differentiating its category of web services. Therefore, it is crucially harmful to affect the classification accuracy. (2) On the other hand, most existing approaches leverage traditional classification algorithm (e.g., SVM), where multiple basic models need to be trained as a whole to perform web service classification, because each of them is a dichotomous classifier that cannot directly solve a multi-class problem. As a result, they accomplish the task with high complexity both on huge space consumption and slow convergence speed, when training a service classifier on a large-scale web service repository.

To handle above two research challenges, this article proposes a novel framework for effective and efficient web service classification. The authors mainly focus on solving two issues: (1) How to enrich short text description of web services towards more accurate correlation with its category? (2) How to implement service classification efficiently? To address the first issue, the authors first eliminate those words (Del-List) from original service descriptions, since they cannot make differences among different service categories. Then, the authors derive keywords (Key-List) that can affect the capability of service topic recognition. Finally, the authors propose an expansion strategy to enrich service functionality description of web services. To address the second issue, the authors train augmented service descriptions by using labeled probabilistic topic model L-LDA which mines the implicit semantic relationships throughout service repository. The authors assign all the services categories as feature topics when training an L-LDA model to generate probabilistic topic distribution. That greatly enhances the accuracy and efficiency of service classification work. The main contributions of the work is as follows:

1. The authors propose a novel framework for web service classification with the consideration of both augmenting original service descriptions and mining latent semantics of service topic features.
2. The authors propose an expansion strategy for augmenting original service descriptions via Word2vec by jointly eliminating words that are occurred among multiple service topic distributions with high probability and deriving keywords that can significantly reflect service category.
3. The authors propose a service classification learning approach based on labeled probabilistic topic distribution model L-LDA by using augmented service descriptions.
4. The authors design and implement a prototype system and conduct extensive experiments on a real-world data set crawled from PrgrammableWeb. The results demonstrate that the approach achieves superior performance than state-of-the-art web service classification methods.

The remainder of the paper is organized as follows. Section 2 reviews the related work. Section 3 formulates the web service classification problem. Section 4 illustrates an overview framework of the proposed approach. Section 5 presents the details of the approach for web service classification. Section 6 shows the experimental evaluation, and Section 7 concludes the paper and discusses the future work

5. RELATED WORK

In recent years, service classification has become one of the core research issues in the field of services computing. In general, service classification approaches fall into two categories: syntactic-based service classification and semantic-based service classification.

Syntactic-based service classification methods focus on applying traditional classification algorithms to differentiate the tags of web services. Simon et al. presented an approach that performed pool-based active learning with support vector machine (SVM) for text classification (Tong & Koller, 2001). The method was based on the motivation that when the labels of the service training set need to be manually labeled, how to efficiently choose unlabeled services at each round and achieve the maximum classification accuracy with the least manual consumption. After that, Wang et al. took advantage of SVM to classify web services into corresponding categories (Wang, Shi, Zhou, Bouguettaya, 2010). During the process of service classification, a standard taxonomy, UNSPSC, was used to model the feature space of web services. Although SVM algorithm performs well in service classification, only a single classification method without any preprocessing procedure still cannot achieve the best classification performance. To recommend proper services, Molood et al. classified web services from the observation of QoS values at different levels. However, QoS-aware classification mainly focuses on different non-functional features of web services that does not consider the differentiations in terms of service functionality (Makhlughian, Hashemi, Rastegari, & Pejman, 2012).

Semantic-based service classification methods leverage latent probabilistic topic model and traditional classification algorithms to recommend tags for web services. SVM algorithm and Formal Concept Analysis (FCA) have been used by Marcello et al. to automatically classify web services to specific domains (Bruno, Canfora, Penta, & Scognamiglio, 2005). The FAC is used to analyze the ontology structure of the service description content, by which the semantic-based service classification is achieved. Krestel et al. (2011) introduced an approach based on LDA model to elicit latent topics from resources. Aznag, Quafafou, and Jarir (2014). extended the correspondence LDA model to automatically tag web services according to existing manual tags by exploiting local correlation labels in multi-label learning context. Katakis, Meditskos, Tsoumakas, Bassiliades, and Vlahavas (2009) improved web service classification accuracy by taking into account both the textual and semantic description, where machine learning algorithms were also leveraged to promote the classification process. To reduce the human effort on labeling services, Liu et al. (2016) proposed a scalable service classification approach by active learning algorithm. They adopted LDA technique to model a service description as a feature vector with latent topics in terms of semantic level, which is used to iteratively learn an optimum SVM model as web service classifier. Shi et al. (2017) proposed a multi-label active learning approach for web service tag recommendation, expanding Liu's work (2016) into a multi-label scenario. Active learning was considered to train a multi-label classifier with a correlation-aware learning strategy, which can learn the correlation relationship among different tags. Liang, Chen, Wu, and Bouguettaya (2016) presented a graph-based approach to automatically assign tags to unlabeled API services by exploiting both graph structure information and semantic similarity. However, most currently existing approaches mainly count on the original service descriptions, which cannot fully reflect its topic features and semantic expansion characteristics for web service classification, leading to low service classification accuracy.

Observed from the above deficiencies, the authors propose a novel approach for web service classification via augmenting labelled probabilistic topic model. By combining the augmentation of original service descriptions using semantic feature extraction, it achieves better service classification accuracy compared to the existing methods, while learning a labelled probabilistic topic model with good convergence speed.

3. PROBLEM FORMULATION

In this section, the authors mainly focus on providing the definitions on authors' research issue. First, the authors abstractly describe the RESTful API service system and mashup service system, respectively. Then, the authors formulate a web service classification problem to be solved.

Definition 1 (API Service Repository). A 4-tuple $ASR = < S, SD, C, f >$ is defined as an API service repository, where

1) $S = \{s_1, s_2, ..., s_m\}$ represents a set of API services in the repository, where m is the number of API services.
2) $SD = \{sd_1, sd_2, ..., sd_m\}$ is a collection of original service descriptions to their API services, where $sd_i = \{w_{i_1}, w_{i_2}, ...\}$ consists of all of the words in a service description.
3) $C = \{c_1, c_2, ..., c_l\}$ represents a set of API service categories. Each of them stands for a kind of service domain.
4) f is a function mapping from an API service to its corresponding categories. That is, $\forall s_i \in S$, there is a service category c_j derived by mapping function $f(s_i)$.

Given an $ASR = < S, SD, C, f >$, it is observed that some of the API services share the same category, each of which includes at least one API service.

Definition 2 (Mashup Service Repository). A 3-tuple $MSR = < M, MD, g >$ is defined as a mashup service repository, where

1) $M = \{m_1, m_2, ..., m_p\}$ represents the set of mashup services in the repository, where p is the number of mashup services.
2) $MD = \{md_1, md_2, ..., md_p\}$ is the collection of original service descriptions to their mashup services, where $md_i = \{w_{i_1}, w_{i_2}, ...\}$ consists of all the words in a mashup service description.

3) g is a function mapping from a mashup service to its corresponding API services. That is, $\forall m_i \in M$, a set of API services $S' \subseteq S$ can be calculated by $g(m_i)$ as the integrated components when creating the mashup service m_i.

Based on the above two definitions, the authors formulate the web service classification problem as below.

Definition 3 (Web Service Classification). A 3-tuple $WSC =< ASR, MSR, r >$ is defined as web service classification problem, where $ASR =< S, SD, C, f >$ is an API service repository, $MSR =< M, MD, g >$ is a mashup service repository, and $r = \{w_{r_1}, w_{r_2}, ...\}$ represents the set of words of a service registration request. A solution of a web service classification problem responds to a service category $c_k \in C$.

Given a web service classification problem $WSC =< ASR, MSR, r >$, the authors propose a novel approach for discovering a service category and describe its procedure in the subsequent section.

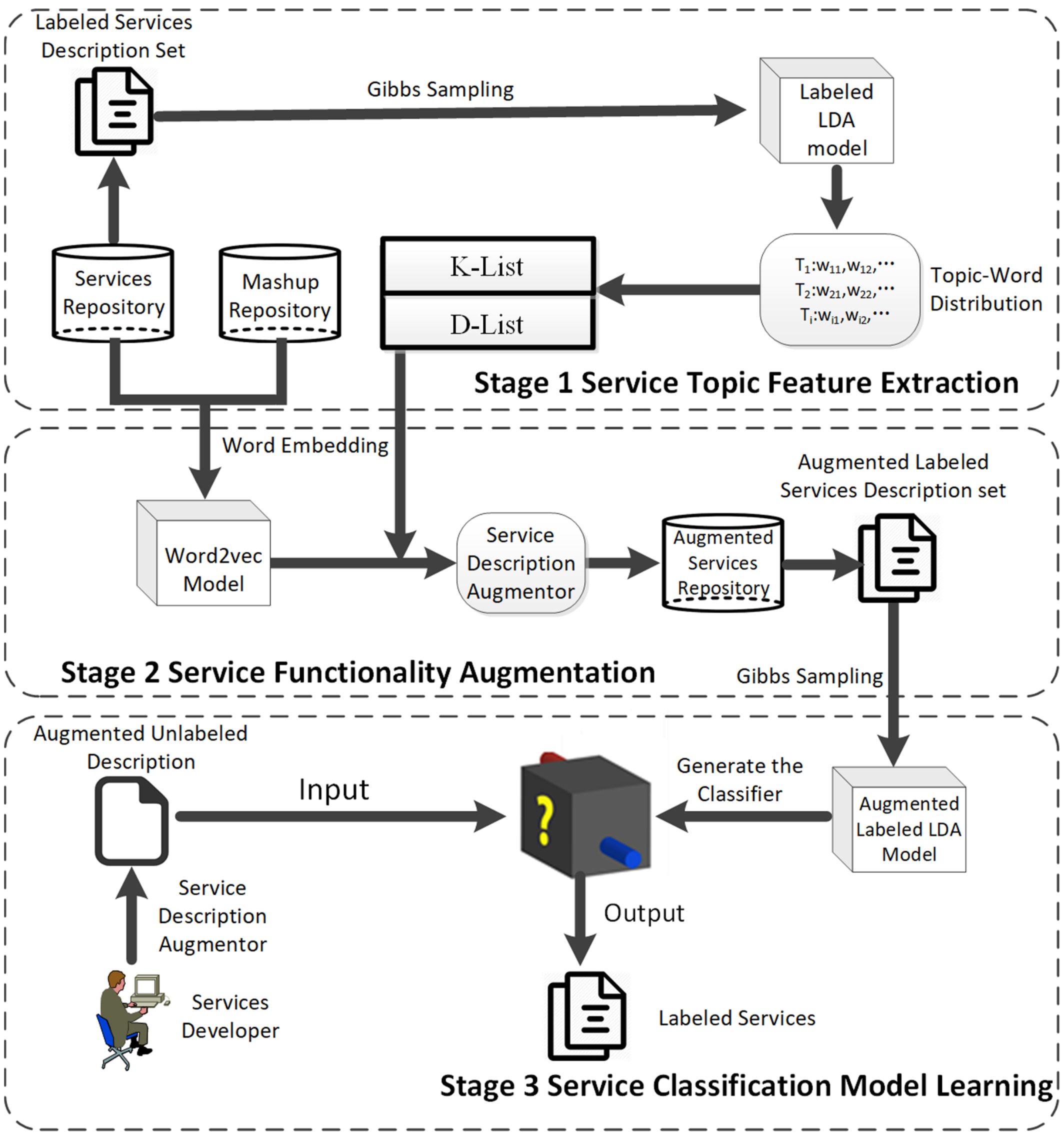
4. FRAMEWORK

Figure 1 illustrates the overall framework of the proposed approach for web service classification. From the perspective of task functionality, it goes through three independent but correlative crucial stages: (1) service topic feature extraction, where the processes are carried out by analyzing the probabilistic distributions of latent topics; (2) service functionality augmentation, where service description reduction and expansion are performed by the techniques of word embedding; (3) service classification model learning, where the procedure of training a service classifier is accomplished via labeled probabilistic topic model.

In the stage of service topic feature extraction, first, multiple categories are mined from the service descriptions based on original web service repository. Second, taking all of these categories as latent topics, a labeled service probabilistic topic distribution model can be learned by training raw web services via Gibbs sampling. Similar to traditional LDA model, the result of the training process is represented as two matrices, including a service text description document-service topic matrix and a service topic-service functionality description word matrix. In the document-topic matrix, each service description corresponds to a probability distribution on different categories. Similarly, in the topic-word matrix, each service category corresponds to a probability distribution on different words. Finally, by analyzing the topic-word matrix, the authors extract a list of redundant words called Del-List and a list of keywords called Key-List as service topic features. Those words that appear with high probability across multiple service categories are recognized as redundant elements (Del-List), since they disturb the purity of service description and reduce the classification accuracy. On the contrary, those words that are derived from the updated topic-word matrix with the consideration of Del-List are recognized as keywords (Key-List), which are top-ranked at each service category distribution and raise the classification accuracy. The results (Del-List and Key-List) obtained from the service feature extraction are fed to the stage of service functionality augmentation.

In the stage of service functionality augmentation, taking the obtained service features as inputs, the authors perform two parallel steps of reduction and expansion for enriching original web service descriptions. During the reduction step, according to the elements of Del-List, the authors make a removal of functionality description on web services. During the expansion step, a word embedding model Word2vec is first learned by training the dataset from collected web services. Through the model, utility function is applied to measure the semantic similarity degree among different words of service descriptions. Then, the authors retain those words from Key-List in the service descriptions and expand them by selecting the closest words based on semantic similarity calculation. By the augmentation of service functionality descriptions, the authors transform the original web service repository to an enriched one, which is fed as training set to the stage of service classification model learning.

Figure 1. The overall framework of the approach for web service classification

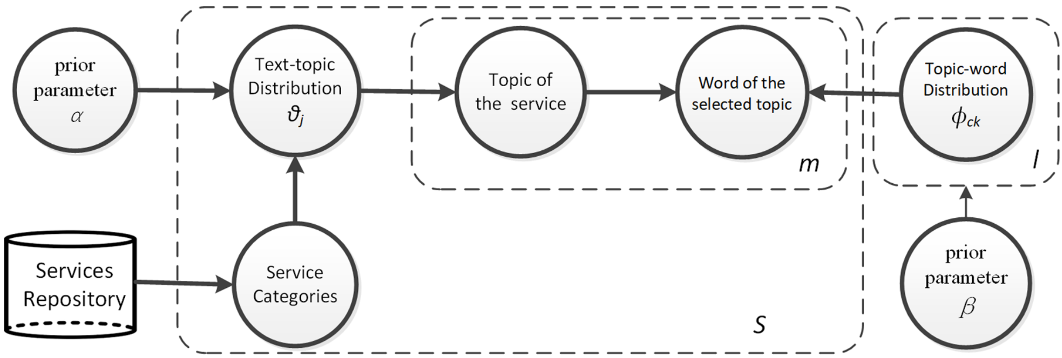


In the stage of service classification model learning, a service classifier is first generated by training augmented web service repository via labeled probabilistic topic model L-LDA. Then, a service functionality description from a service provider to be classified is augmented in the same way. Finally, the authors apply the service classifier to recommend a category, which would assist the provider in publishing the web service to an online service management platform.

5. WEB SERVICE CLASSIFICATION

In terms of the three stages in the framework of web service classification, the authors describe each of them in this section. First, the authors present elaborate to extract the feature of service topics. Second, the authors present how to derive a word embedding model and augment service functionality

Figure 2. Graphical Labeled LDA model of generating service probabilistic topic distribution



description. Finally, the authors describe how to generate the service classifier by training a labeled probabilistic topic model based on the enriched API service repository.

5.1. Service Topic Feature Extraction

In this section, the authors first present generation process of probabilistic topic distribution from an API service repository, based on which the authors analyze the topic-word distribution and extract service topic feature.

5.1.1. Generating Service Probabilistic Topic Distribution

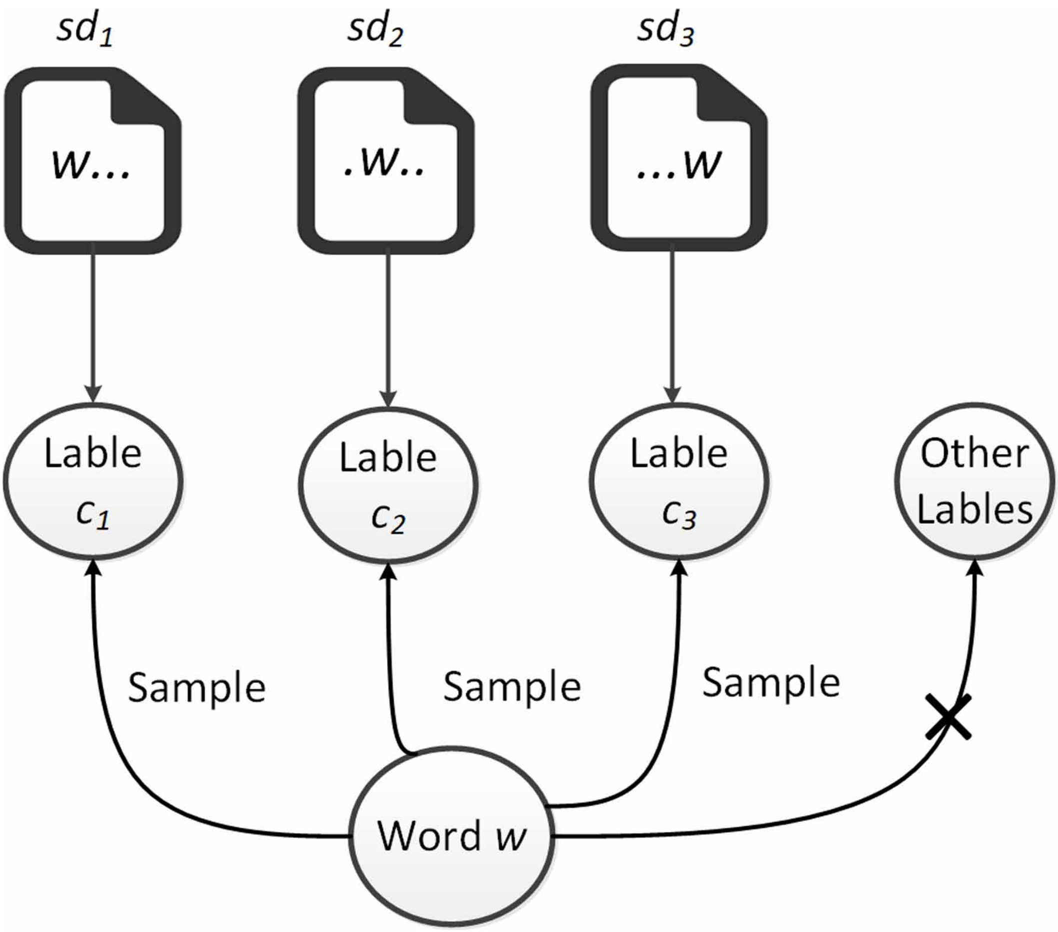
The Labeled Latent Dirichlet Allocation (L-LDA) model (Blei, Ng, & Jordan, 2003) is applied to calculate the topic features of the existing web services. Given an API service repository $ASR = < S, SD, C, f >$, there are a set of web services $S = \{s_1, s_2, ..., s_m\}$, a set of service descriptions $SD = \{sd_1, sd_2, ..., sd_m\}$ and respective service description words $sd_i = \{w_{i_1}, w_{i_2}, ...\}$. The process of generating a word (Ramage, Hall, Nallapati, & Manning, 2009) in service description is illustrated in Figure 2, given topic-word Distribution and text-topic distribution.

As the opposite process, the process of generating service probabilistic topic distribution can be described as below.

For each service topic $c_k \in C \ (1 \le k \le l)$
Draw $\varphi_{c_k} \sim \text{Dirichlet}(\beta)$;
For each API service $s_j \in S$
Draw $\theta_j \sim \text{Dirichlet}(\alpha)$
For each $w \in sd_i$
Draw a service topic $z \sim Multinomial(\theta_j)$
Draw a service description word $w \sim Multinomial(\varphi_z)$

where l is the number of service topics from C and φ_{c_k} is the multinomial distribution over words specific to service topic c_k and θ_j is the multinomial distribution over service topics specific to an API service $s_j \in S$. α and β are the prior parameters of Dirichlet distribution for θ_j and φ_{c_k}, respectively.

Figure 3. Supervised service words sampling process in L-LDA



To generate the service probabilistic topic distribution, it goes through two steps where Gibbs sampling (Heinrich, 2008) is applied to infer the desired parameters θ and φ, including supervised service words sampling and iterative update training.

In the step of supervised service words sampling, all the words from service descriptions in $SD = \{sd_1, sd_2, ..., sd_m\}$ are accumulated together in a unified collection. Since labeled topics correspond to API service categories $C = \{c_1, c_2, ..., c_l\}$, when sampling a service word w that appears in a set of service descriptions $SD' = \{sd_i, sd_j, ..., sd_k\} \subseteq SD$, it is randomly sampled into only a set of restricted topics $C' = \{c_i, c_j, ..., c_k\} \subseteq C$ that are applied to label those service descriptions in SD'. For example, there is a service word w appearing in three service descriptions $SD' = \{sd_1, sd_2, sd_3\}$. The corresponding labels of each service description in SD' are $C' = \{c_1, c_2, c_3\}$, respectively. In such case, The service word w is only randomly sampled into the restricted service categories $C' = \{c_1, c_2, c_3\}$, which is shown in Figure 3.

It is observed that the service words sampling process of training L-LDA has restricted service categories, instead of completely and randomly sampling of traditional LDA. Typically, it is a supervised training process when sampling a service description word into a service category. In this way, the sampling process can reach convergence faster than traditional LDA.

In the step of iterative update training, once completing one round of service words supervised sampling, the Gibbs sampling method is used for iteratively training the L-LDA model. These two steps continue to be alternatively performed until it reaches the convergence condition and outputs the estimated parameters. The iterative update training for service probabilistic topic distribution works by selecting one dimension of the probability vector at a time and sampling the values of the current dimension based on the values of other dimensions. The above update training process is mainly achieved by Gibbs sampling parameter estimation.

$$P\left(C_i = c \middle| \vec{C}_{\neg i}, \vec{w}\right) \propto p\left(C_i = c, w_i = w \middle| \vec{C}_{\neg i}, \vec{w}_{\neg i}\right) = \hat{\theta}_{sc} \cdot \hat{\varphi}_{cw} \tag{1}$$

The above is the process of resampling the topic *c* according to the keyword distribution of other words for *w* after the random sampling topic of which is removed. The resulting $\hat{\theta}_{sc}$ and $\hat{\varphi}_{cw}$ are the parameter estimates of the two Dirichlet posterior distributions on topic-word and service-topic. According to Dirichlet parameter estimation formula, we can get

$$\hat{\theta}_{sc} = \frac{n_{s,\neg i}^{(c)} + \alpha_c}{\sum_{c=1}^{C} (n_s^{(c)} + \alpha_c)}, \hat{\varphi}_{cw} = \frac{n_{c,\neg i}^{(w)} + \beta_w}{\sum_{w=1}^{sd} (n_{c,\neg i}^{(w)} + \beta_w)} \tag{2}$$

In the end, we can get the derived Gibbs sampling update formula as

$$P\left(C_i = c \middle| \vec{C}_{\neg i}, \vec{w}\right) \propto \frac{n_{s,\neg i}^{(c)} + \alpha_c}{\sum_{c=1}^{C} n_s^{(c)} + \alpha_c} * \frac{n_{c,\neg i}^{(w)} + \beta_w}{\sum_{w=1}^{sd} n_{c,\neg i}^{(\boldsymbol{w})} + \beta_w} \tag{3}$$

Here α_c and β_w are hyper parameters of the two Dirichlet posterior distributions. The right side of the formula is actually $p(topic \mid service) * p(word \mid topic)$, which represents a probability path of service → topic → word.

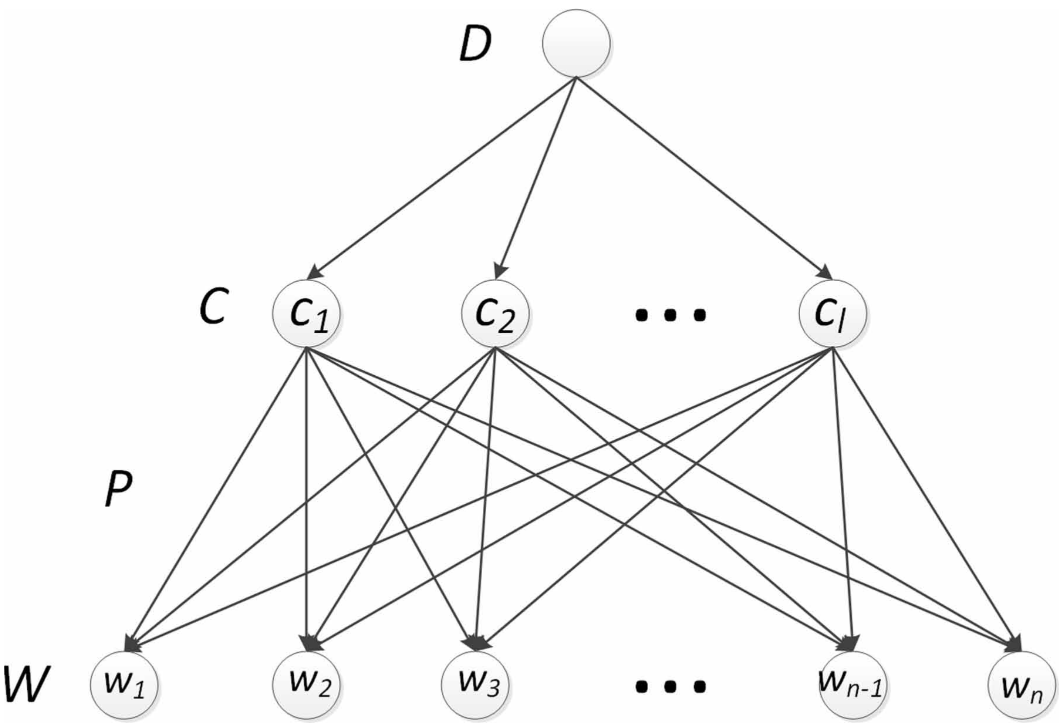
After the training on API service repository by the two steps, the derived L-LDA model generated the estimates of two parameters θ and φ. They can be formulated as the following two matrices.

$$M_{s\_c} = \begin{array}{c} s_1 \\ \vdots \\ s_m \end{array} \begin{pmatrix} p_{11} & \cdots & p_{1l} \\ \vdots & \ddots & \vdots \\ p_{m1} & \cdots & p_{ml} \end{pmatrix}_{m*l} \tag{4}$$

$$M_{c\_w} = \begin{array}{c} c_1 \\ \vdots \\ c_l \end{array} \begin{pmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{l1} & \cdots & p_{ln} \end{pmatrix}_{l*n} \tag{5}$$

where M_{s_c} is the service descriptions and service topics matrix; M_{c_w} is the service topics and service words matrix.

Figure 4. The weighted service topic distribution tree transformed from L-LDA model



5.1.2. Mining Service Topic Feature

After training the original API service repository $ASR =< S, SD, C, f >$, the authors generate a labeled LDA model, where two probabilistic distributions are derived and represented as service-topic matrix M_{s_c} and topic-word matrix M_{c_w}. Especially, service categories $C = \{c_1, c_2, ..., c_l\}$ are assigned as topics during the learning process. To extract topic feature from the learned L-LDA model, the authors transform the service topic-word matrix into a weighted topic distribution tree with three layers, which is defined as below and illustrated in Figure 4.

Definition 4 (Weighted Topic Distribution Tree). Given an API service repository $ASR =< S, SD, C, f >$, the authors transform its learned service topic-word probabilistic distribution matrix M_{c_w} as a three-layer weighted tree, denoted as a 4-tuple $CW\_Tree =< D, C, W, P >$, where

1. D represents the whole service probabilistic topic distribution model located in the first layer of the tree;
2. $C = \{c_1, c_2, ..., c_l\}$ is the collection of all service topics located in the second layer of the tree;
3. $W = \{w_1, w_2, ..., w_n\}$ is the collection of all words across service descriptions located in the three layers of the tree;
4. $P = \{C_1 \rhd P_1, C_2 \rhd P_2, ..., C_l \rhd P_l\}$ is the collection of the probabilistic distribution of service topics C on W, where $\sum_{k=1}^{n} P_i(wk \mid C_i) = 1$.

As illustrated from the weighted topic distribution tree, many leaf nodes have multiple edges from their upper nodes with high probabilities. That is, a service func0074ionality description word has been distributed across varieties of service topics with top-ranked probability. Therefore, these kinds of words cannot reflect the service category characteristics. On the contrary, it decreases the purity of differentiating its topic feature of a web service and affects the classification accuracy. From the above observation, the authors analyze and traverse the weighted service topic distribution tree to extract service topic feature, including a list of redundant words called Del-List and a list of keywords called Key-List.

To extract those redundant service description words, traversing the weighted service topic distribution tree on leaf nodes is performed. When a leaf node $w \in W$ passes through a large number of topics $\{c_i, c_j, ..., c_k\} \subseteq C$ with high weights from $P = \{C_1 > P_1, C_2 > P_2, ..., C_l > P_l\}$, it is extracted as a redundant word and pruned from the tree by deleting the leaf node and its corresponding edges with upper nodes. After the iterations of tree traversing and pruning, the authors can generate all the redundant words in $Del - List = \{w_p, w_q, ...\}$ Del-List and an updated weighted service topic distribution tree.

When extracting the service topic feature on keywords Key-List, the authors traverse the updated weighted service topic distribution tree on service topic nodes in the second layer. For each service topic node $c \in C$, those leaf nodes $\{w_i, w_j, ..., w_k\} \subseteq W$ that are connected with c and the weights of their edges are ranked at Top-K are extracted as its keywords. Finally, the authors generate the $Key - List = \{C_1 \rhd \{w_1^1, w_1^2, ...\}, C_2 \rhd \{w_2^1, w_2^2,, ...\}, ..., C_l \rhd \{w_l^1, w_l^2,, ...\}\}$ by extracting all of the keywords of service topics.

5.1.3. Service Functionality Augmentation

In this section, the authors first describe the process of training a word embedding model, which is then leveraged to augment service functionality description based on the extracted service topic feature.

5.2. Training Word Embedding Model of Service Descriptions

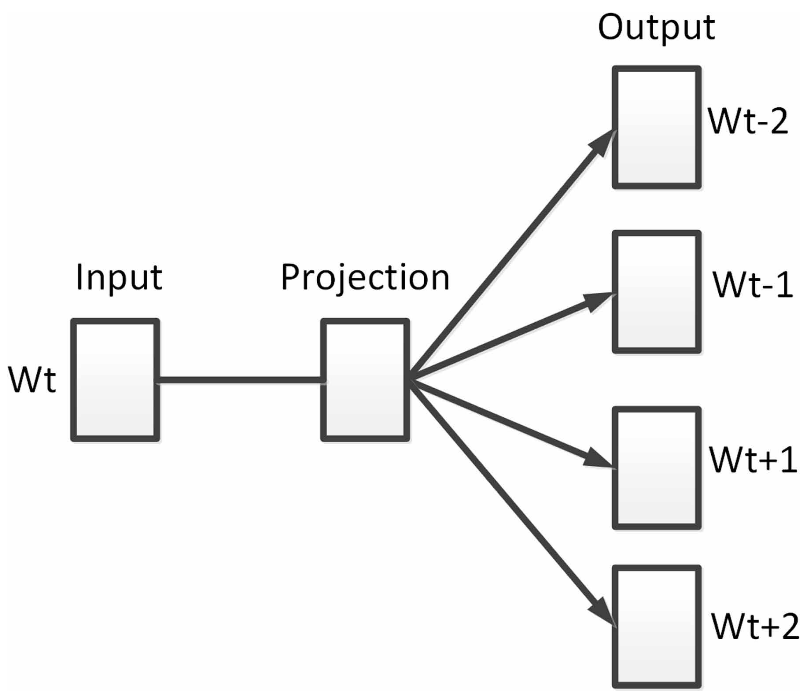
To expand a service word from an API functionality description, the measurement of similarity distance between semantic representation vectors of two words is performed by a trained word embedding model. Specifically, the authors train an integrated service repository including large-scale API services and mashup services crawled from ProgrammableWeb online service registration platform, where more than 18,000 service descriptions are accumulated as Word2vec model training set. After the training, the authors derive a word embedding model called Word2vec (Mikolov, Chen, Corrado, & Dean, 2013).

The goal is to derive an optimum Word2vec neural network model that generates a semantic representation vector for a service description word after finishing the training process. Traditionally, there are two ways to optimize the parameters of Word2vec model, including Word Bag strategy which measures the semantic representation vectors of target words by those words around them and Skip-Gram strategy which measures the semantic representation vectors for the surrounding words starting from the current word.

In this paper, the authors take advantage of Skip-Gram strategy to calculate joint probability between the vector of an input target service word and the vectors of several output service words, which is used to optimize the parameters of the neural network and adjust word vector itself. Figure 5 shows the evaluation of the surrounding words, where the input to Skip-Gram strategy is the semantic representation vector of the current word and the output include the probability of those vectors for the surrounding words. The training process is formulated as below.

$$p(w_o \mid w_t) = \frac{\exp(v'_{w_o} v'_{w_t})}{\sum_{w=1}^{|W|} \exp(v'_w v_{w_t})} \tag{6}$$

Figure 5. The evaluation of surrounding words by Skip-Gram strategy



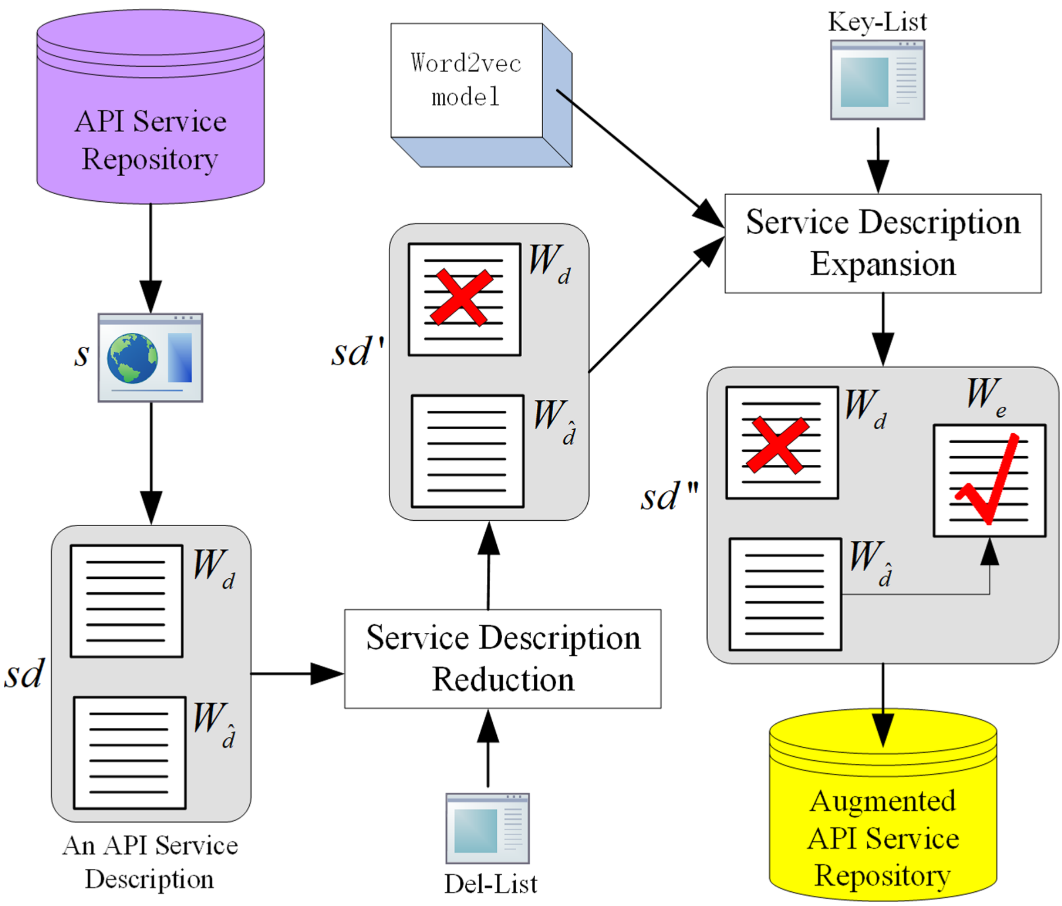
where $p(w_o \mid w_t)$ is the probability of output word w_o given a target word w_t, W is the set of surrounding words specific to w_t, v and v' are the eigenvectors of the input and output words, respectively. By the calculation of the cosine similarity between the word vectors from web service description, the results of the Skip-Gram strategy are fed into neural network for parameter training. Finally, the authors can derive a Word2vec model for the conversion from a word to its corresponding semantic representation vector that is applied for enriching the service functionality description.

5.2.1. Augmenting Service Functionality Description

Given an API service repository $ASR =< S, SD, C, f >$, the authors perform two steps for enriching the service description of each API service, including service description reduction and expansion. The process of augmenting an API service description is illustrated in Figure 6.

During the step of service description reduction, the authors directly remove all of the service words from an API service description, which are matched with the extracted redundant elements $Del - List = \{w_p, w_q, ...\}$. For an API service $s \in S$, its corresponding service description $sd = \{w_i, w_j, ...\}$ includes a set of service description words. The authors partition these service description words into two independent sets, where $sd = W_d \cup W_{\hat{d}}$ and $W_d \cap W_{\hat{d}} = \varnothing$. After the partitioning, all of the words in W_d are subsumed in $Del - List$, then the authors get $W_d \subseteq Del - List$. Meanwhile, all of the words in $W_{\hat{d}}$ have no intersection with $Del - List$, which leads to the result $W_{\hat{d}} \cap Del - List = \varnothing$. Finally, the authors prune all of the elements from W_d and generate a reduced API service s' with its service description sd', where has the same service description words with $W_{\hat{d}}$.

Figure 6. The process of augmenting an API service description



During the step of service description expansion, the authors choose part of those words from generated service description sd' and augment their semantic functionality description. First, the authors select a candidate service description word for expansion when it is satisfied by the authors' enriching strategy. It is defined as below.

Definition 5 (Service Word Expansion Selection). Given a reduced API service s' with its service description sd' and extracted keywords $Key-List = \{C_1 \triangleright \{w_1^1, w_1^2, ...\}, C_2 \triangleright \{w_2^1, w_2^2,, ...\}, ..., C_l \triangleright \{w_l^1, w_l^2,, ...\}\}$, its service category can be calculated by topic mapping function f. Suppose that the authors have $c_k = f(s') \in C$. By matching the service category with $Key-List$, the authors derive a subset of words $W_k = \{w_k^1, w_k^2,, ...\}$, which has a set of service keywords specific to C_k. The condition of service word selection is that, if a service word both satisfy $w_i \in W_{\hat{d}}$ and $w_i \in W_k$, then it is selected to be expanded.

By applying the service expansion selection strategy to each $w_i \in W_{\hat{d}}$, the authors can choose a set of service words included in $w_i \in W_k$, which is denoted as $W_{\hat{d}}^{'} = W_k \cup W_{\hat{d}}$. Through the learned Word2vec model, the authors model each service word from $w_i \in W_{\hat{d}}^{'}$ as word vector. Suppose that a word vector for $w_i \in W_{\hat{d}}^{'}$ is generated and represented as $w_i = \{u_1, u_2, ..., u_m\}$, while a word vector for a candidate word from training set is represented as $w_c = \{v_1, v_2, ..., v_m\}$. Then the distance calculation between two service words w_i and w_c can be performed by

$$Dis(w_i, w_c) = \sqrt{\sum_{r=1}^{m}(u_r - v_r)^2} \tag{7}$$

In terms of distance calculation, the closest Top-K words are chosen as the expanded set for a service word $w_i \in W_d^{'}$. Note that K can be dynamically changed according to the probability of the word distributed in C_k. As a result, all of the chosen words for each service word $w_i \in W_d^{'}$ are accumulated as W_e and added to the service functionality description. By doing so, the reduced service description sd' is expanded to an augmented one sd''.

5.2.2. Service Classification Model Learning

After the augmentation for an original API service repository $ASR = < S, SD, C, f >$, the authors have enriched functionality descriptions of web services, denoted as $ASR' = < S, SD', C, f >$, where each of web service description $sd' \in SD'$ is augmented relevant to $sd \in SD$. Based on the augmented API service repository, the authors learn a labeled probabilistic topic model again by Gibbs sampling, where all of the service categories are also assigned as the labeled topics. The boosted L-LDA model can be further used as one of the components in the service classifier with the support of more precise probabilistic topic distributions.

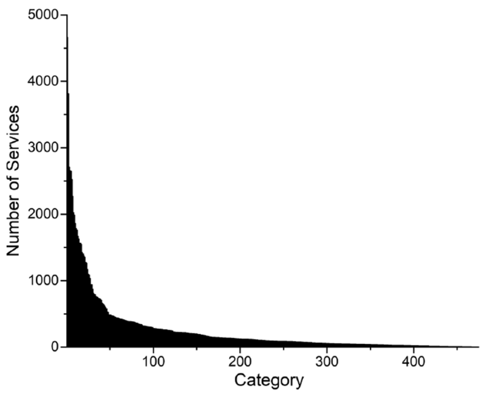
When a service provider submits a service description $r = \{w_{r_1}, w_{r_2}, ...\}$ to be registered on a web service management platform, a service category can be determined with the highest probability that is selected out of a great number of probabilistic topic distributions by applying the learned service classification model. The generated classification is recommended to the service provider for effectively and efficiently completing the service registration process. Compared with the traditional service classification approaches, the prominent advantages of the novel service classifier are listed as below.

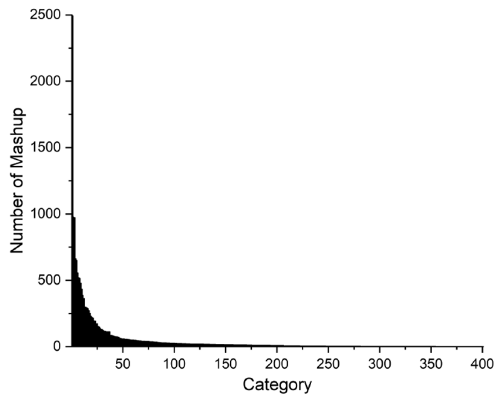
1. Through the strategy of service topic feature extraction, it accurately extracts the crucial features of web services. Moreover, those service description words that frequently appear but without contribution to service classification are filtered out. However, most existing approaches do not consider the reduction of those words, leading to low classification accuracy. By doing so, it provides a basis for service functional description augmentation in API or mashup service repository.
2. Applying the semantic augmentation mechanism on service training set, it can further overcome the shortcoming of insufficient service description. However, most existing approaches still lack of service semantic expansion. As a result, the novel classifier can generate more accurate classification for the service developer.
3. Labeled-LDA as a supervised probabilistic topic model can reach convergence state in a shorter period of time during the training and updating process, compared with other models (e.g., SVM, LDA). The fast response makes it possible and potentially applicable for service developers to exploit real-time service categories for online service registration.

6. EXPERIMENTAL EVALUATION

In this section, the authors conduct a set of experiments on large-scale real service dataset to validate the accuracy and efficiency of the proposed approach for classifying web services. The authors

Figure 7. The Data distributions on category and its corresponding number of web services





mainly prove the performance of the proposed approach in two aspects. First, as the number of the services in training set changes, the authors compare the service classification performance among SVM-based approach, LDA-SVM active learning approach, the proposed traditional L-LDA based approach, and the proposed improved L-LDA based approach. Second, the authors investigate how the setting of those parameters in the approach influences the performance of web service classification.

In experimental datasets, each API service mainly consists of three attributes: ApiName, CategoryName and Service Description, while each mashup service mainly consists of four attributes: MashupName, CategoryName, APIs and functionality Description. The authors use functionality descriptions from all API services and mashup services as a training set to derive the Word2vec embedding model. For service classification model learning, the authors choose 3600 the most informative API services belonged to 8 categories in total, where the number of services for each category ranges from 229 to 827. The selected service categories include Mapping, Social, e-Commerce, Search, Tools, Messaging, Video and Financial. The dataset of the selected web services for service classification learning is shown in Table 1.

6.1. Experimental Settings and Dataset

The authors developed a prototype system in Java. It allows a service developer to submit a service with its registration functionality description as input, and recommends a classification with highest probability as output by implementing two kinds of approaches, including a traditional L-LDA based and an improved L-LDA based web service classification. By doing so, the authors can assist a service provider effectively and efficiently finish the process of service registration on a real-world service management platform, such as ProgrammableWeb. Meanwhile, two existing approaches SVM-based approach (Ames & Naaman, 2007) and LDA-SVM active learning approach (Lopez & Maldonado, 2016) are integrated in the service classification system. The authors ran the experiments on a PC with Intel(R) core(TM) i5-5200U processor 2.20 GHz and 8 G RAM in Microsoft Windows 10 operating system.

The authors designed and implemented a web crawler by Python to collect 13869 real web services and 6254 mashups from ProgrammableWeb (www.programmaleweb.com), which is the largest RESTful online service registration and management platform. There are 400 service categories across all of the API and mashup services. The data distributions on category and its corresponding number of services are illustrated in Figure 7.

Table 1. The dataset of the selected web services for service classification learning

Service category	Mapping	Social	e-Commerce	Search	Tools	Messaging	Video	Financial
Number of services	375	461	496	259	827	490	229	545

Figure 8. The experimental results of service classification among the four approaches

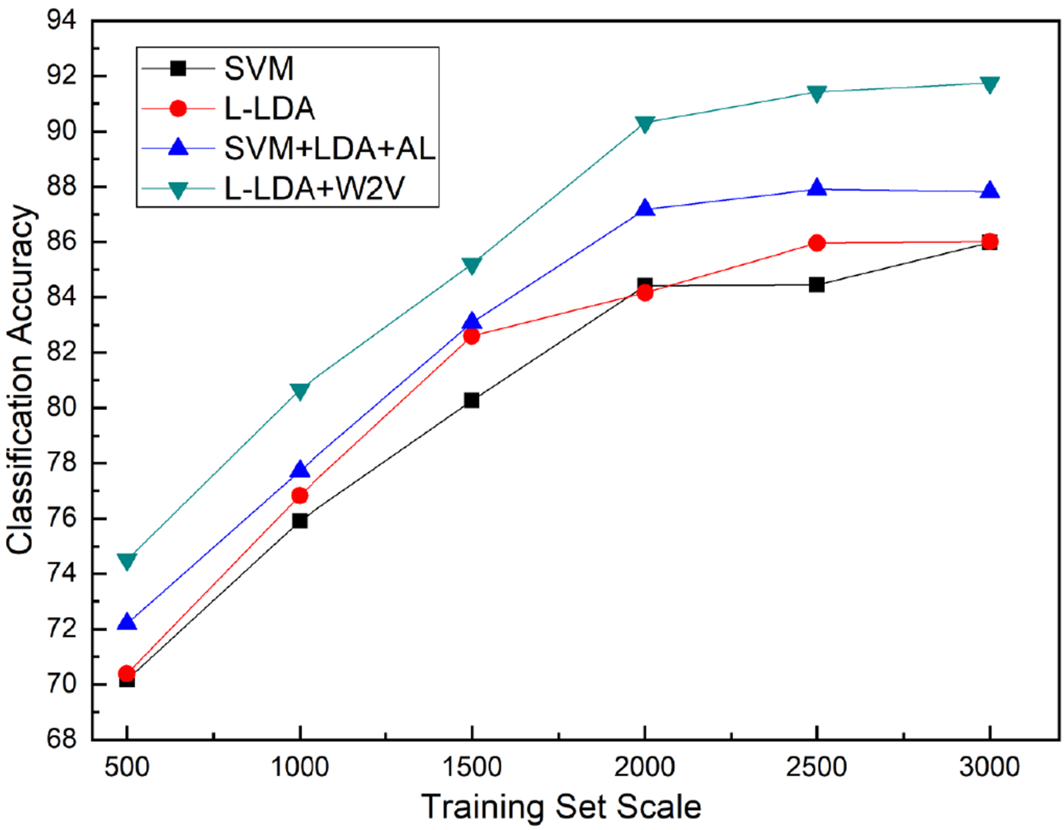


Table 2. The experimental results on each service category of the four approaches

	Mapping	Social	e-Commerce	Search	Tools	Messaging	Video	Financial
SVM	85.21	83.02	85.11	83.63	81.52	90.25	86.32	88.21
L-LDA	85.92	83.18	83.21	84.52	87.25	89.32	88.74	87.88
LDA+SVM	88.09	87.93	86.95	84.16	89.15	89.52	87.47	90.91
L-LDA+Word2vec	91.74	90.09	88.62	91.93	90.74	91.86	92.82	93.49

6.2. Competitive Methods and Parameter Setting

To demonstrate the performance of the proposed traditional L-LDA based and improved L-LDA based service classification methods, the authors compare with another two competitive methods which are related with the work. The service classification of the compared four methods is described as below.

1. SVM-based service classification method. The authors treat each service as a feature vector using the TF-IDF algorithm, where the number of terms can be extracted from all service descriptions in API service repository. Then, the service vectors are fed as the training data to learn the SVM model for service classification.
2. LDA-SVM based service classification method. The authors first use the LDA model to extract the features of web services and transform each service into an L-dimensional vector by probabilistic topic distribution on predefined latent topics, where L is equal to the number of service categories. Then, the authors take the obtained service feature vectors as inputs of training data and apply the active-learning strategy for SVM model learning, which leads to a service classifier.
3. The proposed traditional L-LDA based service classification method. The authors directly take advantage of labeled probabilistic topic model L-LDA as the service classifier, where service categories are fed as latent topics during the training process by the original service descriptions. After learning the L-LDA model as service classifier, the topic with the highest probability is recommended as the service classification result for category registration, when a service provider submits a service functionality description.
4. The proposed improved L-LDA based service classification method. Unlike directly learning a L-LDA model, the authors first augment service functionality descriptions of the original API service repository using word embedding techniques, where it eliminates those words uncorrelated with its corresponding service topic and enriches those words highly reflecting its corresponding service topic. Then, the authors take the augmented service descriptions as inputs and learn a more robust L-LDA model as service classifier.

6.3. Experimental Results and Analysis

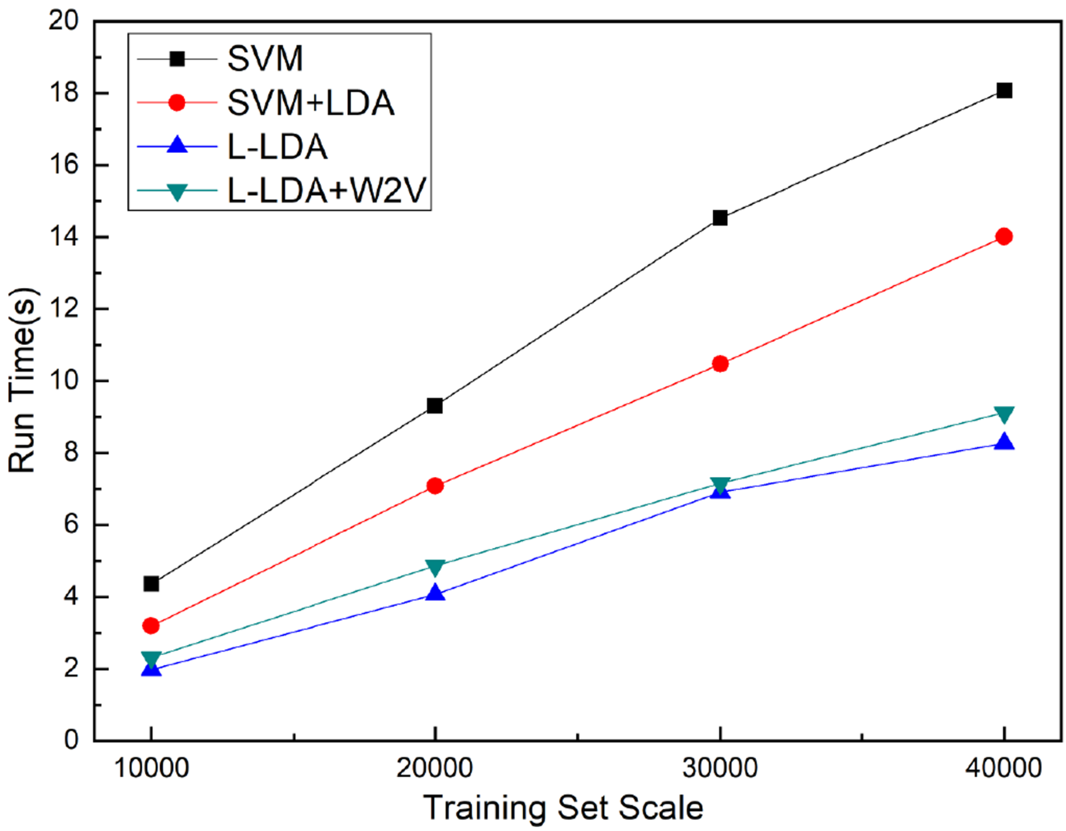
The authors evaluate the service classification performance from the perspective of accuracy and convergence speed. Figure 8 shows the experimental results of service classification among the four approaches, including the two proposed methods and another compared two methods.

Furthermore, the authors also analyze the differences of the classification results on each service category of the four methods when the training set scale is 3000, which is summarized in the Table 2.

It is observed from the above experimental results that the authors' method achieves a superior accuracy of service classification compared to two competitive methods and the self-developed method. SVM-based method achieves comparatively low service classification accuracy as it mainly exploits TF-IDF algorithm which only considers the syntactic level on feature extraction among service descriptions. The proposed traditional L-LDA method outperforms SVM-based method in that it takes semantics into account when extracting service feature vector with labeled topics, where those words that belong to the same service category are semantically gathered into a topic, causing a more accurate service classification result.

Moreover, LDA-SVM based method performs better than the previous two in that it classifies web services with the consideration of combining semantic feature extraction and active learning driven SVM learning. The proposed improved L-LDA method can achieve the best service classification accuracy. It shows two advancements that makes the result superior: First, augmenting service descriptions make the training dataset more accurately recognized for classifier learning; Second, labelled probabilistic topic model is applied to boost the semantic relationships of service description words.

Figure 9. The experimental results on convergence speed among the four approaches



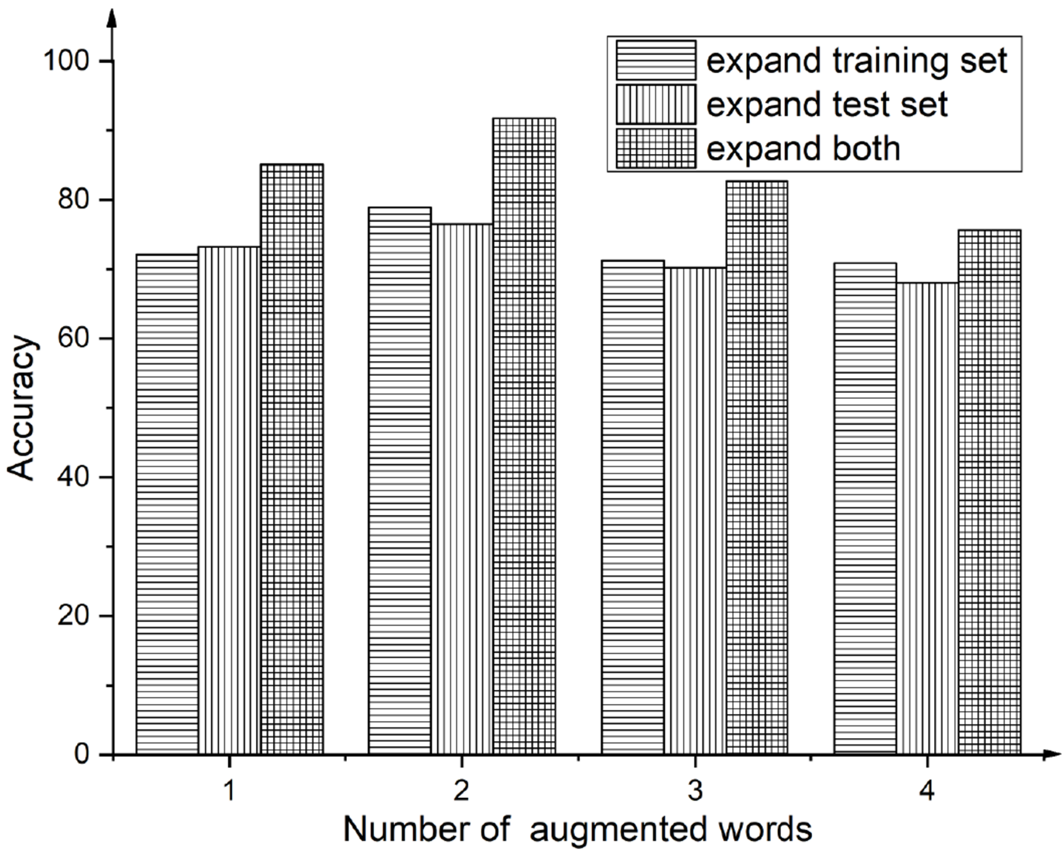
As to the efficiency on different service classification methods, the authors test the convergence speed and the trend of the four training methods as the dataset becomes larger. The experimental results are illustrated in Figure 9. The results prove that the proposed L-LDA based method has the best convergence speed. Meanwhile, as the size of dataset increases, the growth trend of the convergence speed is the slowest among the four service classification methods.

In addition to the experimental results with competitive methods, different parameter settings of the proposed method affect the accuracy of service classification. The authors test the number of augmented words specific to a keyword. Here, the authors fix the number of running iterations of training L-LDA model as 30, and then range the number of augmented words from 1 to 4. Figure 10 shows the experimental results of parameter infiuence of augmented words on classification accuracy. The experimental results are shown from three aspects, including the expansion on the training set only, on the test set only and on both the training set and test set at the same time. From the results, the authors find out when the number of augmented words is set as 2, it tends to reach the best classification accuracy.

7. CONCLUSION

In this paper, the authors propose a novel approach to address the service classification problem. The key idea is that original service descriptions are reconstructed to enrich the semantic feature of web services. It goes through three crucial steps: 1) extracting service topic features, where redundant elements and keywords are respectively detected by modeling and traversing a three-layer weighted

Figure 10. The parameter influence of augmented words on classification accuracy



topic distribution tree; 2) augmenting service functionality descriptions, where a Word2vec word embedding model is learned to perform keyword expansion and redundant elements are pruned based on the extracted service topic features; 3) learning service classification model, where labeled probabilistic topic model is exploited to train a service classifier leveraging the augmented API service repository. Comprehensive experiments conducted on original web services from ProgrammableWeb validate the effectiveness and efficiency of the proposed approach.

In future work, the authors will extend this work on multi-label service classification. They explore how to accurately recommend multiple categories that are more likely used on a real-world service management platform, where service providers are required to submit their registration by selecting a sequence of service categories.

ACKNOWLEDGMENT

This work was supported by Shanghai Natural Science Foundation (18ZR1414400 and 17ZR1400200), National Natural Science Foundation of China (61772128), National Key Research and Development Program of China (2017YFC0907505) and Fundamental Research Funds for the Central Universities (16D111208). Guobing Zou served as the corresponding author during the development and writing of this article.

REFERENCES

Ames, M., & Naaman, M. (2007). Why we tag: motivations for annotation in mobile and online media. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 971–980). ACM. doi:10.1145/1240624.1240772

Aznag, M., Quafafou, M., & Jarir, Z. (2014). Multilabel learning for automatic web services tagging. *International Journal of Advanced Computer Science & Applications*, *5*(8). doi:10.14569/IJACSA.2014.050827

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research Archive*, *3*, 993–1022.

Bruno, M., Canfora, G., Penta, M. D., & Scognamiglio, R. (2005). An Approach to support Web Service Classification and Annotation. In *Proceedings of the IEEE International Conference on E-Technology, E-Commerce and E-Service* (pp. 138-143). IEEE. doi:10.1109/EEE.2005.31

Heinrich, G. (2008). *Parameter estimation for text analysis.* Technical Report.

Katakis, I., Meditskos, G., Tsoumakas, G., Bassiliades, N., & Vlahavas, I. (2009). On the combination of textual and semantic descriptions for automated semantic web service classification. In *Proceedings of the IFIP International Conference on Artificial Intelligence Applications and Innovations* (pp. 95–104). Boston: Springer. doi:10.1007/978-1-4419-0221-4_13

Krestel, R., Fankhauser, P., & Nejdl, W. (2009). Latent dirichlet allocation for tag recommendation. In *Proceedings of the third ACM conference on Recommender systems* (pp. 61–68). ACM.

L'opez, J., & Maldonado, S. (2016). Multi-class second-order cone programming support vector machines. *Information Sciences*, *330*, 328–341. doi:10.1016/j.ins.2015.10.016

Li, J., Luo, X., Xia, Y. N., Han, Y. K., & Zhu, Q. S. (2015). A time series and reduction-based model for modeling and QoS prediction of service compositions. *Concurrency and Computation*, *27*(1), 146–163. doi:10.1002/cpe.3208

Liang, T. T., Chen, L., Wu, J., & Bouguettaya, A. (2016). Exploiting heterogeneous information for tag recommendation in api management. In *Proceedings of IEEE International Conference on Web Services* (pp. 436–443). IEEE. doi:10.1109/ICWS.2016.63

Liu, X. M., Agarwal, S., Ding, C., & Yu, Q. (2016). An lda-svm active learning framework for web service classification. In *Proceedings of the IEEE International Conference on Web Services* (pp. 49–56). IEEE. doi:10.1109/ICWS.2016.16

Makhlughian, M., Hashemi, S. M., Rastegari, Y., & Pejman, E. (2012). Web service selection based on ranking of QoS using associative classification. *International Journal on Web Service Computing*, *3*(1), 1–14. doi:10.5121/ijwsc.2012.3101

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *Computer Science*, *3*(7).

Ramage, D., Hall, D., Nallapati, R., & Manning, C. D. (2009). Labeled lda: a supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 248–256). EMNLP. doi:10.3115/1699510.1699543

Shi, W. S., Liu, X. M., & Yu, Q. (2017). Correlation-aware multi-label active learning for web service tag recommendation. In *Proceedings of the IEEE International Conference on Web Services* (pp. 229–236). IEEE. doi:10.1109/ICWS.2017.37

Tong, S., & Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, *2*(1), 45–66.

Tsoumakas, G., Katakis, I., & Taniar, D. (2008). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, *3*(3), 1–13. doi:10.4018/jdwm.2007070101

Wang, H. B., Shi, Y. Q., Zhou, X., & Bouguettaya, A. (2010). Web service classification using support vector machine. In *Proceedings of IEEE International Conference on TOOLS with Artificial Intelligence* (pp. 3–6). IEEE. doi:10.1109/ICTAI.2010.9

Xia, Y. N., Liu, Y., Liu, J., & Zhu, Q. S. (2012). Modeling and Performance Evaluation of BPEL Processes: A Stochastic-Petri-Net-Based Approach. *IEEE Trans. Systems, Man, and Cybernetics*. *Part A*, *42*(2), 503–510.

Xia, Y. N., Luo, X., Li, J., & Zhu, Q. S. (2013). A Petri-Net-Based Approach to Reliability Determination of Ontology-Based Service Compositions. *IEEE Transactions on Systems, Man, and Cybernetics. Systems*, *43*(5), 1240–1247. doi:10.1109/TSMCA.2012.2227957

Shengye Pang is currently a third-year master student in the School of Computer Engineering and Science, Shanghai University, Shanghai, China. He received his Bachelor's Degree in the College of Electronics and Information Engineering, Tongji University, Shanghai, China. His research interests include web service classification, service clustering and QoS-based service prediction and recommendation. He has published a paper on International Conference on Service Oriented Computing.

Guobing Zou is an Associate Professor in the School of Computer Engineering and Science, Shanghai University, China. He received his PhD in Computer Science from Tongji University, Shanghai, China, 2012. He has worked as a Visiting Scholar in the Department of Computer Science and Engineering at Washington University in St. Louis from 2009 to 2011, USA. His current research interests mainly focus on services computing, data mining, intelligent algorithms and recommender systems. He has published around 60 papers on premier international journals and conferences, including IEEE Transactions on Services Computing, IEEE International Conference on Web Services, International Conference on Service Oriented Computing, IEEE International Conference on Services Computing, International Journal of Web and Grid Services, AAAI, Information Sciences, Expert Systems with Applications, Knowledge-Based Systems, Applied Intelligence, Journal of Internet Technology, Journal of Web Semantics and Soft Computing. He served as an organization chair and program committee member on several international conferences. He worked as a reviewer for IEEE Transactions on Services Computing, Journal of Artificial Intelligence Research, and IEEE Transactions on Systems, Man and Cybernetics: Systems.

Yanglan Gan is an associate professor in the school of Computer Science and Technology, Donghua University, Shanghai, China. She received her Ph.D. degree in computer science from Tongji University, Shanghai, China, 2012. Her research interests include Bioinformatics, web services and data mining. She has published more than 30 papers on premier international journals and conferences, including Bioinformatics, BMC Bioinformatics, IEEE/ACM Transactions on Computational Biology and Bioinformatics, IEEE International Conference on Web Services, International Conference on Service Oriented Computing, IEEE International Conference on Services Computing, Neurocomputing and Knowledge-Based Systems. She served as a program committee member on varieties of international conferences. She worked as a reviewer for BMC Bioinformatics, Knowledge-Based Systems, APBC and ADMA. She is the corresponding author on this paper.

Sen Niu received a Ph.D. degree from the School of Computer Engineering and Science, Shanghai University, China. He is currently working as an Assistant Professor at Shanghai Polytechnic University, Shanghai, China. His research interests include web service composition, uncertain QoS and automated planning. He has published around 10 papers on international journals and conferences, including IEEE International Conference on Web Services, IEEE International Conference on Services Computing, International Conference on Service Oriented Computing, Journal of Internet Technology, International Journal of Web and Grid Services and International Journal of Web Engineering and Technology.

Bofeng Zhang is a full professor in the School of Computer Engineering and Science, Shanghai University, China. He received his Ph.D. degree from the Northwestern Polytechnic University (NPU), China, 1997. He experienced a Postdoctoral Research at Zhejiang University from 1997 to 1999, China. He worked as a visiting professor at the University of Aizu from 2006 to 2007, Japan. He worked as a visiting professor at Purdue University from 2014 to 2015. His research interests include artificial intelligence, personalized service recommendation, intelligent human-computer interaction and data mining. He has published more than 150 papers on international journals and conferences. He worked as the program committee chair for UUMA from 2011 to 2016. He also served as a program committee member for multiple international conferences.