International Journal of Multimedia and Ubiquitous Engineering
Vol.11, No.12 (2016), pp.433-442
http://dx.doi.org/10.14257/ijmue.2016.11.12.39

Service Clustering by Leveraging a Context-Sensitive Approach

Lantian Guo[1], Tao Yang[1, 2], Huixiang Zhang[1], Dejun Mu[1], Zhe Li[1] and Yang Li[1]

[1]School of Automation, Northwestern Polytechnical University, Xi'an, Shaanxi, China 710072
[2]State Key Lab for Manufacturing Systems Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi, China 710049
rcglt@163.com

Abstract

Service technology has gained increasing popularity in recent communication software applied in many domains. With a growing number of services that share same or similar functionalities, clustering services help improve both service composition and mashup creation. To achieve service clustering, utilizing probabilistic topic model to extract and characterize the service description documents as corresponding topics is an available scheme. However, unlike short text in social networks, the descriptions of published services possess higher dimensionality and sparse functional information. With traditional LDA (Latent Dirichlet Allocation) model to implement topic extraction makes topics unclear. To address that challenge, we conduct a context sensitive approach to generate context sensitive vector for merging the words with similar context before loading to LDA model, referred to as CV-LDA (Context Vector LDA). Through F1-Measure of clustering and topic perplexity analysis in the real-world dataset, it is shown that the proposed approach outperforms traditional LDA model in service clustering.

Keywords: *Service clustering, short text, topic model, context sensitives*

1. Introduction

With the explosive development of Internet of Things (IoT) and Social Networks, humans and Smart things can communicate and interact together via service platforms incorporating the IoT and Social Networks. Service Oriented Architecture (SOA) is a widely-used middleware model linking different functional units through defined interfaces between these services [6]. With the SOA techniques, both individual and enterprises can develop, publish web or cloud services, and apply them to commercial information systems or personal applications.

The rapid increasing number of internet services provide the same or similar functionalities, it is a challenge for service users to select a suitable service. To address service discovery and clustering issues, many researchers proposed text mining approaches for clustering, since many service providers can publish service description on their websites [7].

In order to classify different documents, LDA (Latent Dirichlet Allocation) model is proposed to detect their hidden topics from a large scale documents. Specifically, LDA a probabilistic graphical model which can learn the topic representation of each document and the words associated to each topic [2]. With hidden topics, different documents can be classified and clustered. LDA model has many successful applications on news articles and academic articles abstracts. However, unlike these, service description corpus is extreme short and sparse, which is so high dimensionality incredibly that hinders model learning precision.

Word embedding is a word vectorization technique in natural language processing where words can be mapped to vectors in a low-dimensional space. Word2vec is a set of

ISSN: 1975-0080 IJMUE

models to generate neural word embedding vectors relying on a skip-grams model [11]. This model predicts source contextual words from the target words by setting a word windows from which one word is excluded. This context-sensitive approach can capture semantic context features in the corpus, which can be used to cluster the words with similar context features for representing semantic efficiently and reducing the dimensionality of LDA model.

To achieve service clustering, utilizing probabilistic topic model to extract and characterize the service description documents as their corresponding topics is an available scheme. In this paper, we conduct a context sensitive based method for service description topic and service clustering. Its main contributions are as follows:

1) Introduce limitations of LDA model in service description corpus processing assignment, and the superiority in context sensitive vectorization approach.

2) Implement a service clustering method utilizing a context sensitive approach. More important, we design a similarity based context word merging algorithm for incorporating context sensitive approach into LDA model.

3) Crawl a large number of latest service description corpus in real-word, and get experiment results from this real-world dataset.

2. Related Work

A lot of existing service discovery and clustering approaches concentrate on collected QoS (Quality of Service) rating and ranking information of services [4,15], however, the truth is service developers have only published limited QoS information. And for a user, it is not possible to test all services to get QoS information.

To address that limitation, some service discovery and clustering approaches limited to keyword-based framework by matching on names, locations, businesses and defined characteristic in the service description [14]. However, it is difficult for users to be aware of the suitable and correct keywords to retrieve the targeted services. And it is a hard work for developer to define and structure the description file. For example the WSDL(Web Services Description Language)is a kind of typical structured service description format.

The keyword-based service discovery and clustering approaches cannot make use of heterogeneous and unstructured semantic information. To handle the drawbacks of the keyword-based methods, some text mining techniques are applied to extract service features. Services can be clustered utilizing the clustering or classification after features extraction. Chen, *et al.* [3] use LDA topic model to extract feature word in WSDL file of service description. Kumara, *et al.* [8] proposed a hybrid of ontology based approach to calculating semantic similarity of services. It utilized ontology-learning method to seek the hidden semantic patterns among services descriptions, and employ WordNet as measurement tool of the similarity.

Aznag, *et al.* [1] implement several probabilistic topic models: Correlated Topic Model (CTM), Probabilistic Latent Semantic Analysis (PLSA), and latent Dirichlet allocation(LDA) to extract latent topic in service descriptions. However, most work are basic re-implement of these probabilistic topic models.

Niu, *et al.* [12] compare the semantic feature extracting result of LDA and word embedding respectively. Their experiment result show that since the high dimensionality problem, LDA model show unfavorable performance in semantic feature extracting. Nevertheless, word embedding possesses a better performance than LDA in high dimensionality problem.

As an application of word embedding, Le, *et al.* [9] proposed a Doc2vec method that utilizing word embedding technique to generate vector representation for a document, so that documents can be classified by vector representations. However, for corpus with the high dimensionality and sparse functional information, this method is hard to get a satisfied feature extracting ability.

3. CV-LDA Based Service Clustering

3.1. Overview

Figure 1 shows the framework of our context sensitive service clustering approach. The approach mainly includes the following steps:

1) data set preparation. crawl service description from API website. Then, make the preprocessing to get the initial data set.

2) context sensitive word vectorization and word merging, context sensitive word vectorization for texts in dataset are implemented. After getting similarity of vectors, words with high similarity are merged to achieve the dimensionality reduction.

3) topic-oriented service clustering. merged result is loaded into LDA topic model to train. Every service description can be represented as a probability value term of topic. Clustering algorithms can be implemented to get the service clusters utilizing the probability value terms.

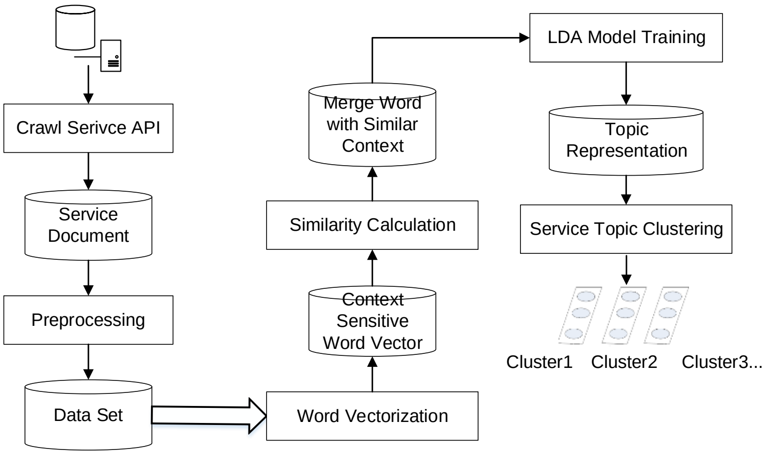


Figure 1. An Overview of Context Sensitive Service Clustering

3.2. Context Sensitive Word Vectorizations

In word embedding algorithms family, Word2vec is one of famous series based on neural network learning. The demo was created by Google. Word2vec provides an efficient implementation of a skip-gram model aiming to calculate vector representations of words [5].

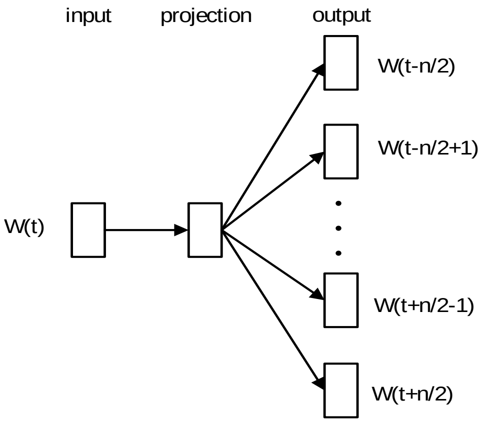


Figure 2. Skip-Gram Model

Figure 2 shows overall structure of Skip-gram. It generates two distinct input vector and output vector to each word. Specifically, it models word co-occurrences extracted within the predefined context window size, through the input and output vectors [11]. Fundamental Skip-gram model define the conditional probability $\mathrm{p}(\mathrm{w}_{\mathrm{t+j}} \mid \mathrm{w}_{\mathrm{t}})$ is:

$$\mathrm{p}(\mathrm{w}_{\mathrm{t+j}} \mid \mathrm{w}_{\mathrm{t}}) = \frac{\exp(\mathrm{v}'^{T}_{\mathrm{w(t+}j)} \mathrm{v}_{\mathrm{w(t)}})}{\sum_{w=1}^{W} \exp(\mathrm{v}'^{T}_{\mathrm{w}} \mathrm{v}_{\mathrm{w(t)}})} \qquad (1)$$

where $\mathrm{v_w}$ and v'_{w} represent input and output word vector of word w among corpus W.

The training sample is a set of one target word and its corresponding context words. Thus, training process can be represented as a group of input and output pairs. Specifically, given a word $\mathrm{w_i}$ in crawled text, training objective is to predict joint probability $p(\mathrm{w_i} \mid \mathrm{Context}(\mathrm{w_i}))$ of $\mathrm{Context}(\mathrm{w_i})$ for word $\mathrm{w_i}$, the formula can be presented as :

$$p(\mathrm{w_i} \mid \mathrm{Context}(\mathrm{w_i})) = \prod_{k=1}^{K} p(d_k \mid \mathrm{q_k}, C) = \prod_{k=1}^{K} (\sigma(\mathrm{q_k} \cdot C)^{1-d_k} \cdot (1-\sigma(\mathrm{q_k} \cdot C)^{d_k})) \qquad (2)$$

where $\sigma(\mathrm{q_k} \cdot C)^{1-d_k}$ and $1-\sigma(\mathrm{q_k} \cdot C)^{d_k}$ present distance between $\mathrm{w_i}$ and its context words in the huffman tree. $\sigma(\mathrm{x}) = 1/(1+\mathrm{e}^{-\mathrm{x}})$ is Sigmoid function. K is path length in the huffman tree. $\mathrm{q_k}$ is non-leaf nodes word vector in huffman branch which objective word $\mathrm{w_i}$ belong to. $\mathrm{d_k}$ is a classification label.

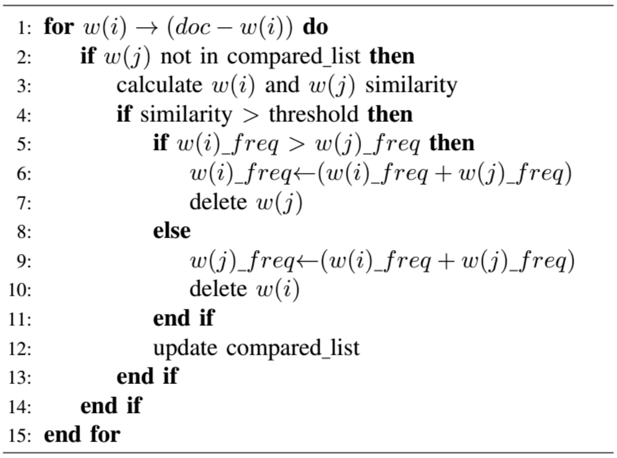
This objective is maximized when the model assigns high probabilities to the real words, and low probabilities to noise words. Then, negative sampling can be utilized to find the solution. After that, utilizing gradient descent to get optimized solution for joint probability in object function, then word vector which has maximum probability value in the Huffman tree can be obtained.

A well trained set of word vectors will place similar words close to each other in that space. For instance, the words "oak", "elm" and "birch" might cluster in one corner, while "war", conflict and strife huddle together in another. Similar things and ideas are shown to be "close". Their relative meanings have been translated to measurable distances.

3.3. Word Merging

Through training process presented above, all of words can be represented as a N-dimension vector. The dimension number "N" was set before training. After vectorization, Euclidean distance is employed to measure similarity between every word pairs.

Before topic modeling process, a bag of word matrix should be built to represent the feature of whole corpus. In order to reduce the dimension of bag of word matrix, we proposed a word merging algorithm, which aims to merge similar words presented in the bag of word matrix. In the merging process, two similar word are decided to merge or not merged on the basis of their similarity. Its pseudocode is shown in Figure 3. $w(\mathrm{i})$ and $w(\mathrm{j})$ are two words in corpus. When $w(\mathrm{i})$ is taken as a target word, and $w(\mathrm{j})$ as a certain word in rest of corpus, the similarity between $w(\mathrm{i})$ and $w(\mathrm{j})$ is calculated and compared with threshold. After that, we merge the word frequency of words with similar context, At the same time, deleting word with lower frequency to update the bag of word matrix.



```
for $w(i) \to (doc - w(i))$ do
    if $w(j)$ not in compared_list then
        calculate $w(i)$ and $w(j)$ similarity
        if similarity $>$ threshold then
            if $w(i)\_freq > w(j)\_freq$ then
                $w(i)\_freq \leftarrow (w(i)\_freq + w(j)\_freq)$
                delete $w(j)$
            else
                $w(j)\_freq \leftarrow (w(i)\_freq + w(j)\_freq)$
                delete $w(i)$
            end if
            update compared_list
        end if
    end if
end for
```

Figure 3. Word Merging Algorithm Pseudocode

3.4. Topic-oriented Service Clustering

After word vectorization and merging in service description, the proposed CV-LDA approach employ LDA model to generate latent topic representation for service description. LDA model is a hierarchical Bayesian model for learning a distribution of documents and words, which attempt to represent the underlying latent structure of the document [3].

LDA can be described as seeking the corresponding latent topics for each service description according to the corresponding distributions over the extracted terms. LDA assigns to the topics for each service description together with the corresponding probability values that the service description contains the corresponding topics [10]. So group those services with similar functionalities into the corresponding topic clusters, consequently to achieve topic-oriented service clustering.

Comparing with classical LDA based approach, our method can merge words with similar context to achieve context sensitive and dimensionality reduction purposes. As shown in Figure 4. Hierarchical model of CV-LDA approach is composed of four layers: service, topic, merged word and traditional word. According to context similarity merged word layer merges the traditional word and updates the bag of word matrix for topic layer.

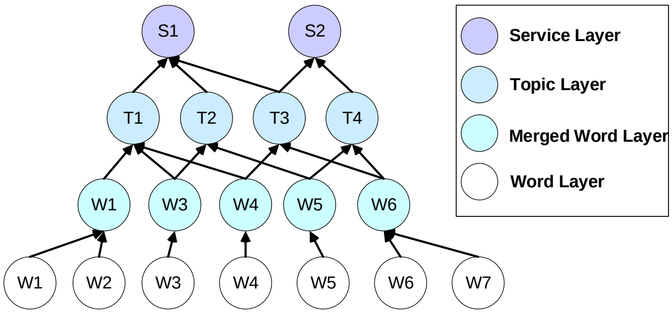


Figure 4. Service Description Representation by CV-LDA

The training process of LDA model is to generate two latent variables: the topic distribution of document θ and the word distribution of topic φ can be estimated from gibbs samples by below:

$$\theta_{m,k} = \frac{n_m^k + \alpha}{\sum_{k=1}^{K} n_m^k + \mathrm{K}\alpha}, \phi_{k,w} = \frac{n_k^w + \beta}{\sum_{w=1}^{V} n_k^w + \mathrm{V}\beta} \tag{3}$$

where n_k^w represent the number of tokens of word w assigned to topic z. n_m^k is the number of tokens in document m are assigned to topic z. α, β is hyperparameters [13].

Overview, the different of sampling result between CV-LDA and traditional LDA is: n_k^w and n_m^k value are updated, since CV-LDA merged word before loading them into LDA model. With the topic distribution of document θ, topic representation of every service description can be generated. Then, we can cluster topic representation of services by utilizing clustering algorithm.

4. Experiment

4.1. Experiment Setup

We crawl service description on the Programmableweb, obtaining description of 10340 active services until Nov. 2015 (http://www.programmableweb.com/). The corpus is preprocessed by removing stop words, stemming and lemmatization. Through preprocessing, we receive a service description corpus containing 852708 words totally. The number of unique token is 27664.

For the context sensitive words vectorization, we employ a Word2vec 0.8 package released by Google in Jul. 2015. Skip-gram model can be implemented by this package. Implementation of LDA model and Gibbs sampling rely on the Gensim python package. Experiment parameters are shown in table 1, where K is the latent topic number which is adjusted in the comparison experiment; α, is set to 50/k and β is set to 0.01 that both of them are empiric values in LDA model training; Number of Gibbs Sampling is set to 300, the reason is that generally, 300 times of iteration can be convergence for LDA model.

Table 1. Service Description Representation by CV-LDA

Parameter	Meaning	Value
Size	Vector Dimensionality	100
Window	Context Window	10
α	Hyper Parameters in LDA	50/K
β	Hyper Parameters in LDA	0.01
Gibbs	Number of Gibbs Sampling	300

4.2. Evaluation Metric

Perplexity is widely adopted to evaluate the performance of topic modeling. It shows how well a probability distribution or probability model predicts a sample. Thus, we utilize perplexity to evaluate topic modeling ability of CV-LDA. The formula as follow:

$$Perplexity(\mathrm{W}) = \exp\left\{ -\frac{\sum_{m=1}^{M} \log(\mathrm{p}(\mathrm{w}_{\mathrm{m}}))}{\sum_{m=1}^{M} N_m} \right\} \tag{4}$$

where w_{m} is observable word in testing document m . $\mathrm{p}(\mathrm{w}_{\mathrm{m}})$ is probability that model can generate w_{m} . N_m is the number of words in document m .

To evaluate the clustering performance, we employ the F1-Measure to show the better performance of CV-LDA. The formula as follow:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{5}$$

4.3. Performance

Table 2. Perplexity Comparison

Number of Topic	CV-LDA	LDA
10	379	584
15	354	567
20	324	550
25	319	550
30	302	539
35	296	528
40	289	550
45	283	533
50	283	561

In out experiment, we utilized all of the crawled service corpus to generate the context sensitive word vectors and implement word merging algorithm. In order to show visualized result, we take 100 services from 5 popular categories (Mapping, Social, eCommerce, Search, Mobile) randomly to train in CV-LDA model.

Since our work is to improve the word layer in LDA model, and LDA model has been widely used in document classification domain, we compare CV-LDA with traditional LDA model in this paper. In future work, we will compare it with more classification algorithms.

Table 2. shows perplexity comparison. The similarity threshold in this experiment is 0.1. It can be found that CV-LDA method outperforms the baseline method in term of perplexity. The lower perplexity, the better in topic modeling ability.

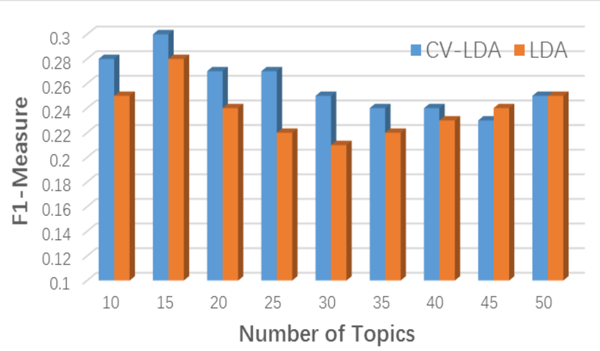


Figure 5. F1-Measure Comparison

Figure 5 shows the F1-Measure comparison in bar chart. After getting topic representation, we employ K-means algorithm to cluster the topic representation. Based on that, the service clustering can be achieved. Since we choose 5 categories in LDA training part, the output of CV-LDA should be clustered into 5 clusters. F1-Measure shows the performance in service clustering. It can be found that, CV-LDA performs better in most case.

Table 3. F1-Measure in Different Similarity Thresholds

Number of Topics / Similarity Thresholds	10	20	30	40
0.05	0.27	0.26	0.23	0.23
0.1	0.28	0.27	0.25	0.24
0.2	0.28	0.25	0.23	0.22
0.3	0.27	0.25	0.22	0.19

Table 3. shows F1-Measure values in different similarity thresholds. When the similarity threshold is 0.1, CV-LDA model shows better performance. We can conclude that either merging too many words or too few words cannot obtain a good performance.it can be obviously found that as the increasing of number of topics in CV-LDA, the service clustering performance become lower. These experiment results can help us choose an appropriate number of topics.

5. Conclusion and Future Works

In this paper, we conduct a context sensitive LDA modeling method for service clustering, referred to as CV-LDA, which utilizes a neural word embedding method to generate context sensitive vector and merge words to achieve dimensionality reduction. Experiment conducted over real-world data shows our method have a lower perplexity and a higher F1-measure.

In future work, we will develop a more powerful method to achieve the dimensionality reduction for extracting high quality topics. And we will conduct more comparisons with existing classification algorithms.

Acknowledgments

We would like to thank anonymous reviewers for their constructive comments. This work was supported in part by the Natural Science Foundation of China (NSFC) under Grant 61672433, and NSFC under Grant 61402373.

References

[1] M. Aznag, M. Quafafou, E. M. Rochd, and Z. Jarir, "Probabilistic topic models for web services clustering and discovery", Proceedings of 3rd European Conference on Service-Oriented and Cloud Computing, **(2013)**, pp. 19-33.

[2] D. M. Blei, "Probabilistic topic models. Communications of the ACM", vol. 55, no. 4, **(2012)**, pp. 77-84.

[3] L. Chen, Y. Wang, Q. Yu, Z. Zheng, and J. Wu, "Wt-lda: user tagging augmented lda for web service clustering", Proceedings of 3rd European Conference on Service-Oriented and Cloud Computing, **(2013)**, pp. 162-176.

[4] X. Chen, Z. Zheng, and M. R. Lyu, "QoS-aware Web Service Recommendation via Collaborative Filtering", Web services foundations, Springer New York, **(2014)**, pp. 563-588.

[5] Y. Goldberg and O. Levy, "word2vec explained: Deriving mikolov negative-sampling word-embedding method", ArXiv preprint, arXiv:1402.3722, **(2014)**.

[6] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, "Interacting with the soa-based internet of things: Discovery, query, selection, and on-demand provisioning of web services", IEEE Transactions on Services Computing, vol. 3, no. 3, **(2010)**, pp. 223-235.

[7] L. Guo, X. Zheng, C. Ding, D. Mu, and Z. Li, "Cloud service recommendation: State of the art and research challenges", Proceedings of the 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, **(2015)**, pp. 761-764.

[8] B. T. Kumara, I. Paik, and W. Chen, "Web-service clustering with a hybrid of ontology learning and information-retrieval-based term similarity", Proceedings of the 20th IEEE International Conference on Web Services, **(2013)**, pp. 340-347.

[9] Q. Le and T. Mikolov, "Distributed representations of sentences and documents", Proceedings of the 31st International Conference on Machine Learning, **(2014)**.

[10] Z. Li, "An on-demand service discovery approach based on mined domain knowledge", Proceedings of the 8th IEEE World Congress on Services, vol. 2012, **(2012)**, pp. 393-396.

[11] P. Liu, X. Qiu, and X. Huang, "Learning context-sensitive word embeddings with neural tensor skip-gram model", Proceedings of the 24th International Conference on Artificial Intelligence, AAAI Press, **(2015)**, pp. 1284-1290.

[12] L. Q. Niu and X. Y. Dai, "Topic2vec: Learning distributed representations of topics", ArXiv preprint, arXiv:1506.08422, **(2015)**.

[13] Z. Qiu, B. Wu, B. Wang, C. Shi, and L. Yu, "Collapsed gibbs sampling for latent dirichlet allocation on spark", Journal Machine Learning Research, vol. 36, **(2014)**, pp. 17-28.

[14] J. Wu, L. Chen, Z. Zheng, M. R. Lyu, and Z. Wu, "Clustering web services to facilitate service discovery", Knowledge and information systems, vol. 38, no. 1, **(2014)**, pp. 207-229.

[15] Q. Yu, "Cloudrec: A framework for personalized service recommendation in the cloud", Knowledge and Information Systems, vol. 43, no. 2, **(2014)**.