2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)

A Web Services Classification Method based on GCN

Hongfan Ye[1], Buqing Cao[1,*], Junjie Chen[1], Jianxun Liu[1], Yiping Wen[1], Jinjun Chen[1,2]

[1]School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China
[2]Swinburne Data Science Research Institute, Swinburne University of Technology, Melbourne, Australia

***ABSTRACT*—With the development of Web2.0 and Web services, the diversity and amount of Web services are increasing quickly. Finding Web services to meet the needs of users has turn into increasingly difficult. It is a valid way to promote service discovery that classifying Web services with similar functionality. The existing Web services classification technology mainly focuses on exploiting the functional information, such as description texts or tags to achieve Web services classification, but ignores the network structure information implied between the words inside the Web service description text and the Web service description text itself. Therefore, we put forward an approach of Web service classification on top of graph convolutional neural networks. This method, firstly, uses the name, description text, tags of Web service as the basic corpus to construct a heterogeneous graph network of "Words & Web service description documents" according to word co-occurrence and word relationship among the Web service description document. In the heterogeneous graph network, all weights of the edges between the document nodes and the word nodes are calculated by used the term frequency-inverse document frequency, and the point mutual information is utilized to calculate the weights of the edges between different word nodes. Then, it applies the graph convolutional neural network to learn the embedding information of words and Web service description document, and transforms Web service documents classification problem into node classification problem. Finally, the experiment on ProgrammableWeb dataset display that the precision, recall, F-measure, purity and entropy of the proposed approach are greatly improved, compared to those of TF-IDF+LR, LDA, WE-LDA, LSTM, Wide&Deep, Bi-LSTM, Wide&Bi-LSTM.**

Keywords- Graph convolution neural network; heterogeneous graph network; WSC-GCN; Web Services classification

INTRODUCTION

With the advent of Web2.0, new information technologies represented by service computing [1], cloud computing [2] and big data [3-5] have been developed rapidly. As a most technology in Service-oriented Architecture (SOA), Web services have also made great progress. Even though Web services published on the network is increasing gradually, these Web services are difficult to meet people's complex functional requirement. Web services are online application services published by organizations to perform certain functions and be designed online to support the interaction between interoperable machines on the network with low cost and high efficiency [6]. There are many Web service description languages, for example, WSDL [7], OWL-S [8], RESTful structured API services, and so on. It has become a challenge problem to immediately and exactly discover the user-desired Web services, and it also becomes more difficult for software developers to discover and reuse service resources effectively. According to the survey, it is very essential to perform the management of Web services through the technologies of service discovery [9], service composition [10]. The classification of Web services with similar functionality is an important approach to support service management, which can extremely decrease the search time and the scope for Web services, thus promote Web service discovery and composition [11].

To enhance Web service discovery, some scholars proposed many Web service classification methods [11, 13]. Most of them address on Web service classification and recommendation on account of functional features [12, 15-18]. The existing researches show that the Web service's function description texts usually have the features with sparse characteristics, short length, and less information, which can be regarded as short text [19]. Therefore, how to construct short texts into a form which can be understood by computers becomes the primary issue for short text classification. To resolve these problems, some researchers exploit the core features derived from WSDL documents to achieve functional classification of Web services [11]. The key feature vectors of each Web service are extracted from the WSDL document. Then, the similarity between the extracted Web service feature vectors are calculated. Finally, according to the similarities of feature vectors of Web service, these Web services are classified as groups with similar functionality. Besides, the latent topic is extracted from Web services by using the LDA [20] topic model or its advanced topic model [13, 21]. Then, the similarity among Web services are calculated based on these topic vectors and the Web Services are classified into various clusters. Recently, it has become one of the research hotspots that the deep mining of the latent information among Web service description texts [14, 22-23]. To summarize, these works greatly facilitate Web Service classification. However, they do not take into account the network structure information implied between the description texts and the words within the description texts, which could be used to enhance service classification.

Therefore, to solve the above limitation, we put forward a Web service classification approach on top of an enhanced graph convolutional neural networks (GCN) in this paper. Firstly, it takes the words and Web service description documents as a single node using a complete corpus based on the real dataset from ProgrammableWeb. Then a "Words & Web service description documents" heterogeneous graph based on these nodes is built and modeled by a GCN [24], which is a simple and efficient graph neural network. The GCN can catch the high-order information of neighborhoods. In the "Words & Web service description documents" heterogeneous graph, word co-occurrence information is

978-1-7281-4328-6/19/$31.00 ©2019 IEEE

DOI 10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00158

used to build the edge between any two-word nodes, and the frequencies of word and document are exploited to construct the edge between word node and document node. Consequently, the process of Web service classification is changed into the process of node classification problem in the heterogeneous graph network. The method can gain a good performance for Web service classification by using a slight number of labeled documents to learn the explicable embedding of word nodes and document nodes. Generally, the main works in our paper are outlined as below:

- We put forward a novel graph neural network model (called as WSC-GCN) for Web service classification. For all we know, this is the first study using Web service dataset to model a whole corpus as a "words & Web service description documents" heterogeneous graph and learn the embeddings of words and Web service description documents with graph neural networks jointly.
- The WSC-GCN is exploited to model and forecast service characteristics of the functionality description documents of Web services. The network structure information hidden between the Web service description document itself and the words appearing in the Web service description document is deeply mined and predicted. The predictions are combined and conformed as the terminal outcome of service classification.
- To verify the proposed method, experimental comparison and analysis in this paper are carried out based on the real dataset from ProgrammableWeb. The experimental outcomes show that our method outperforms the existing methods in the accuracy of Web services classification by comparing with other seven approaches, i.e., TF-IDF+LR, LDA, WE-LDA, LSTM, Wide&Deep, Bi-LSTM, Wide&Bi-LSTM separately. It is worth noting that WSC-GCN also could learn predictive word and document embeddings without exploiting pre-trained word embeddings.

The remainder of our paper is performed as below: we present the related work in Section II. In Section III, the model and method are designed with details. The experimental outcomes and analysis are shown in Section IV. At last, we summarize our work in Section V.

II. RELATED WORK

In an open and varied network environment, a variety of Web services have formed a growing and dynamic service space, which leads to the selection of Web services hard. Therefore, the discovery of Web services has grown a popular topic which has attracted large numbers of eyes of scholars. Some researchers have proved that Web service classification is a productive way to enhance the capability of Web service discovery [11]. Many works related on the classification or clustering of Web services mainly address on the functionality attributes of Web services.

The functionality-based Web service classification is to assort Web services into different clusters according to the function similarity of Web services. For instance, to enhance the quality of Web services classification, some scholars have put forward many ameliorated algorithms for exploiting the semantics in WSDL documents [11]. First of all, the key features from WSDL documents can be exploited, such as the name, messages, content, ports, WSDL, and so on. Then, by using the cosine similarity method or other methods to calculate the semantic similarity between key features, the Web services are divided into similar groups [11]. Through integrating the corresponding tags of description documents and Mashup service, Cao et al. [25] presented an improved K-means algorithm to cluster service. Based on the word vector, Shi et al. [21] proposed an improved service clustering strategy, which incorporates word clustering information into the LDA model training by clustering all words in the Web service description documents. To better model Web service description documents and mine the hidden information, for example, the context and the word order, Shi et al. [22] exploited LSTM to fully mine the functional description of Web service documents. Based on LSTM model, Wang et al. [23] demonstrated a QoS prediction method for learning and predicting the subsequent reliability of service system. Ye et al. [14] presented a Wide & Bi-LSTM combined with linear regression algorithm and Bi-LSTM neural network for short text modeling of Web service function description, to improve the quality of Web services classification. From these research results, it can be seen that extracting more accurate implicit semantic information of services is very helpful for promoting Web service classification. However, the structure graph or network can be built from the real-world Web service datasets. Graph is a special data format, which can be used to represent all kinds of network information (including Web service structure information) in the real world. In a graph, nodes (called "entities") are connected by edges (called "relationships"). The nodes (or edges) in the graph can be mapped to points in vector space through deep learning technology, and then the nodes in vector space can be fully excavated for classification. However, few studies have paid attention to apply graph convolution neural network to Web service data in order to fully mine the network structure information implied between the description documents itself and the words of the Web service description documents so far, which can enhance the quality of Web service classification.

Recently, Yao et al. [26] exploited graph convolutional networks for text classification. Inspired by this research work, a Web service classification approach based on graph convolutional neural network is presented in this paper. The method takes words contained in Web services and its description documents as single nodes, then constructs a "Words & Web service description documents" heterogeneous graph network based on these nodes. As for the "Words & Web service description documents" heterogeneous graph network, we exploit the graph convolutional neural network to learn the embedding information of words and Web service description documents and perform convolution calculation to accomplish Web services classification.

III. WEB SERVICES CLASSIFICATION

The whole framework of the presented approach in our paper is demonstrated in the below Figure 1, which includes three parts. Among this, the first part is the preprocessing of Web service description documents, the second part is the construction and training of WSC-GCN model, and the last part is the Web service classification.

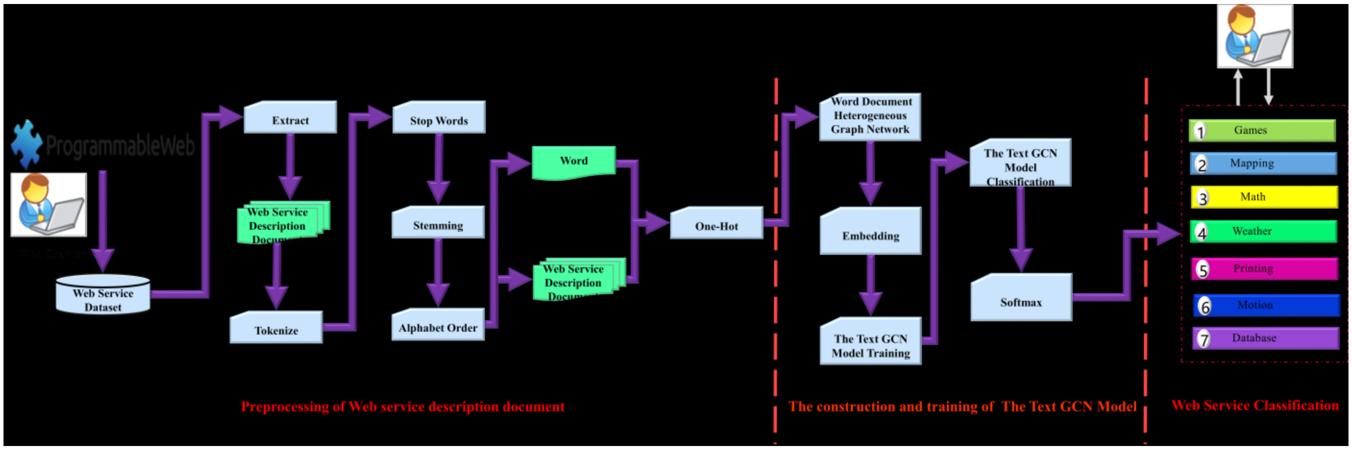

Figure 1. Overall Framework of Web Service Classification

A. Preprocessing of Web Service Description Document

Web service description is the main source information for the classification of Web service. Because some entries of the Web service description documents include lots of useless details, the preprocessing is required to purify the description documents of Web service. The preprocessing involves the below:

Step 1: The data acquisition and organization of Web service description documents. We utilize the NLTK of panda in python to individually draw the five columns of Web APIs, that are *APIName, tags, desc, primary_category, sub_primary*.

Step 2: The tokenization process for Web service description documents. The NLTK also is used to segment words according to spaces, and punctuation marks.

Step 3: Remove stop words. Many invalid words and symbols exist in English, for instance, 'an', 'for', ';', etc. These words and symbols which have no practical meaning are called stop words.

Step 4: Stemming. Depending on the distinct of the tense and person, there are may have different expressions about the same word in English. For instance, 'play', 'playing' and 'plays', are effectively alike the word 'play'. The accuracy of similarity calculation will be dropped, if we deal with these words as different words.

Step 5: Extract the words of the preprocessed Web service description, then perform dictionary processing.

Step 6: Represent the preprocessed Web service description document and each word as a one-hot vector, and construct feature matrices with these one-hot vectors as the input of the WSC-GCN model.

B. The WSC-GCN Model

The built WSC-GCN model is presented in Figure 2, which incorporates three sections: "Words & Web service description documents" heterogeneous graph network, "Words & Web service description documents" representation and Web service classification.

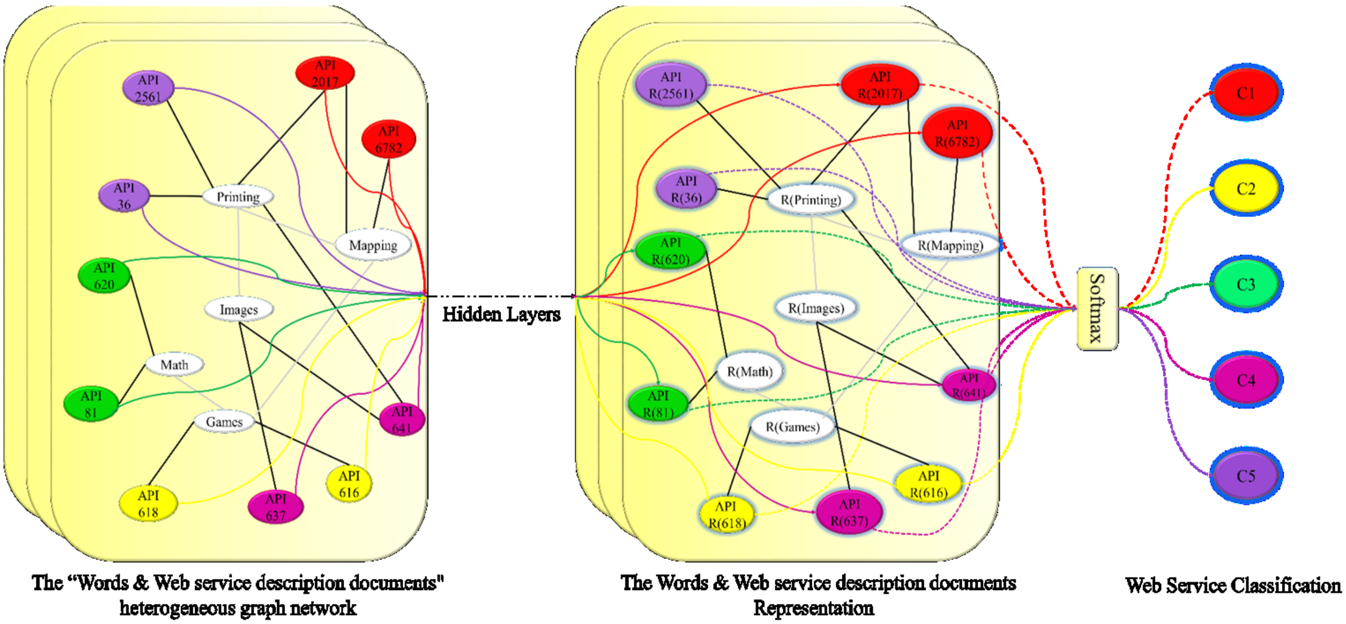

Figure 2. The WSC-GCN Model

1) The Definition of Graph Convolutional Networks

Kipf et al. [37] proposed a graph convolutional neural network (GCN) in 2017. The GCN is a multilayer neural network, which is a variant of the traditional convolution algorithm on the graph structure data. It can be used directly to deal with graph structure data and derives embedding vectors of the nodes according to the features of their neighborhoods. Its definition is as follows:

a) Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $u, v \in \mathcal{V}$ and $(u, v) \in \mathcal{E}$ respectively represent the collection of nodes and edges in the graph.
b) Suppose each node is connected to itself, that is, for any node u has $(u, u) \in \mathcal{E}$.
c) Let $\mathcal{X} \in \mathbb{S}^{n \times m}$ be an m-dimensional matrix including all features of n nodes, where the dimension of the eigenvector is m, the eigenvector of each row for v is $\mathcal{X}_v \in \mathbb{S}^m$.
d) Let A be an adjacency matrix of the graph. Considering the reason for recursion, the diagonal elements of A are all set to 1. In this way, GCN can only use one layer of convolution to catch the information about its nearest neighbors.
e) Let D be the degree matrix of a graph, where $D_{ii} = \sum_j A_{ij}$.

2) The "Words & Web service description documents" Heterogeneous Graph Network

As shown in the left part of Figure 2, we build a heterogeneous graph network which contains word nodes and Web service description document nodes. In the heterogeneous graph network, document node is the node labeled with "API" and its number, and the word node is the other nodes. All document-word edges are labeled with black bold and all word-word edges are labeled with gray thin. The number of nodes in the "Words & Web service description documents" heterogeneous graph network $|\mathcal{V}|$ is the sum of Web service description documents (the corpus size) and the words (vocabulary number) after de-duplication. The edges between nodes are constructed according to the co-occurrence of words from the Web service description documents (denoted as word- document edges) and the co-occurrence of words in the entire corpus (denoted as word-word edges) at the same time. The term frequency-inverse document frequency (TF-IDF) is utilized to calculate the weight of the edge between a document node and a word node. If a word has a high TF in the web service description documents, and it is rarely present in other web service description documents (IDF is also high), the word is with good discrimination ability among different categories. In order to better utilize the global co-occurrence information of words, we use a fixed-size gliding window to collect the co-occurrence statistics of words for all Web service description documents in the corpus. We exploit point-wise mutual information (PMI) to compute the weight between two different word nodes, which is a widespread tolerance of word association. Therefore, we define the weight of the edge between any node i and node j in the V as below:

$$A(i,j) = \begin{cases} PMI(i,j) & i,j \text{ are words}, PMI(i,j) > 0 \\ TF-IDF_{i,j} & i \text{ is document}, j \text{ is word} \\ 1 & i = j \\ 0 & otherwise \end{cases} \quad (1)$$

Next, we calculate the PMI value of a word pair i, j as below:

$$PMI(i,j) = log \frac{p(i,j)}{p(i)p(j)} \quad (2)$$

$$p(i,j)=\frac{\#W(i,j)}{\#W} \quad (3)$$

$$p=\frac{\#W(i)}{\#W} \quad (4)$$

Where $\#W(i)$ is the amount of sliding windows which includes word i in a corpus, and $\#W(i,j)$ is the amount of gliding windows which includes both word i and word j, and $\#W$ is the entire amount of gliding windows in the corpus. A positive PMI value means that the semantic correlation of words in the corpus is high, while a negative PMI value implies that the semantic correlation in the corpus is very small or not at all. Thus, the edges between word pairs with positive PMI values are added only.

3) *The Classified Convolution Calculation for Web services*

After building the "Words & Web service description documents" heterogeneous graph network, we feed it into a simple two layers GCN for modeling and convolutional operations to form the embedding representation vectors of the words and Web service description documents (as shown in the middle part of Figure 2, $R(x)$ means the embedding representation vectors of x). The specific process is as follows:

(1) For a single-layer GCN, the k -dimensional eigenmatrix $L^{(1)}\in\mathbb{S}^{n\times k}$ of a node is calculated as below:

$$L^{(1)}=\rho\left(\tilde{A}\mathcal{X}W_0\right) \quad (5)$$

Where $\tilde{A}=D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ represents the normalized symmetric adjacency matrix, $W_0\in\mathbb{S}^{n\times k}$ represents the weight matrix, and ρ represents the activation function. That is to say, $ReLU\ \rho(x)=max(0,x)$. As mentioned above, stacking multiple GCN layers will integrate more neighborhood information, so that higher order neighborhood information can be obtained:

$$L^{(j+1)}=\rho(\tilde{A}L^{(j)}W_j) \quad (6)$$

Where j denotes the layer number and $L^{(0)}=\mathcal{X}$.

(2) The size of node embeddings (word/document) in the second layer is same to the size of the labels set, and then the node embeddings are fed into a $softmax$ classifier for calculation:

$$\mathbb{Z}=softmax(\tilde{A}\,\mathrm{ReLU}\left(\tilde{A}\mathcal{X}W_0\right)W_1) \quad (7)$$

Where $\tilde{A}=D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ is the same as in equation 1, and $softmax(x_i)=\frac{1}{\mathbb{E}}exp(x_i)$ with $\mathbb{E}=\sum_i exp(x_i)$. The weight parameters W_0 and W_1 can be trained via gradient descent. Thus, if $E_1=\tilde{A}\mathcal{X}W_0$, $E_2=\tilde{A}\,\mathrm{ReLU}\left(\tilde{A}\mathcal{X}W_0\right)W_1$, then E_1 and E_2 can contain the first layer document and word embeddings and the second layer Web service description documents and words embeddings respectively.

(3) The cross-entropy deviation of all labeled Web service description documents is used to define the loss function as follows:

$$\mathcal{L}=-\sum_{d\in y_D}\sum_{f=1}^{F}Y_{df}\,ln\,\mathbb{Z}_{df} \quad (8)$$

Where y_D is the collection of the Web service description documents with labels, and F is the dimension of the output attribute which is equivalent to the number of classes, and Y is the matrix of label indicator.

Therefore, the final Web service classification can be obtained by the convolution calculation of the above two layers of GCN. As shown in the right part of Figure 2, different colors mean different Web service categories. In the "Words & Web service description documents" heterogeneous graph network, it is important to point out that the two-layer GCN allows messages to be passed between nodes, even if there is no direct construction of the connection edges between Web service description documents. As shown in Figure 3, different Web service description documents establish communication links through jointly connected words, so that the Web service description document pairs can exchange information through the jointly connected word nodes, and then carry out classified convolution calculation, so as to guarantee the integrity and consistency of the information.

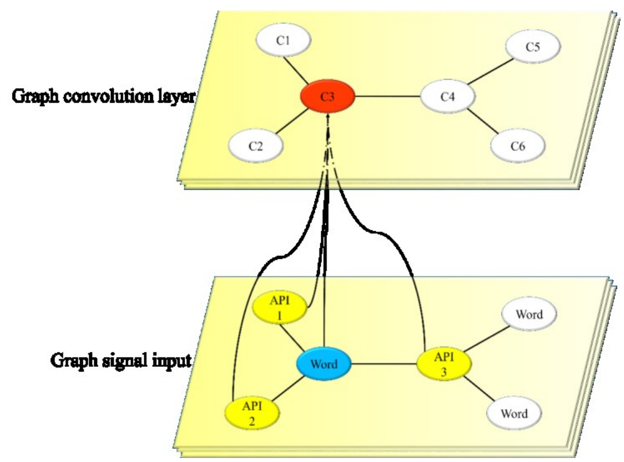

Figure 3. Information Exchange between Web Service Description Documents Pairs

IV. EXPERIMENT

A. *Dataset and Experiment Settings*

The Web service dataset includes 6673 Mashups, 9121 Web APIs, and 13613 links between Web APIs and Mashup, coupled with the Web service description and their tags information, has been crawled from ProgrammableWeb.com. The 9121 Web APIs are selected as experimental dataset, in which the top-10, 20, 30, 40 and 50 Web service categories are chosen as the classification benchmark category. By adopting the random partitioning tool Sklearn, Web APIs in each classification benchmark category are split into 70% training set and 30% test set. Some important parameters in the WSC-GCN model include: Learning_rate=0.02, Epochs=20, Hidden1=20, Dropout=0.5 and Loss weight=0.

B. Evaluation Metrics

We use five indicators to measure the classification quality, which are respectively precision, recall, F-measure, purity and entropy. We assume a standardized classification result s as $SWSC = \{SC_1, SC_2, \ldots, SC_K\}$ for the preprocessed Web service documents set, and the experimental classification result as $EWSC = \{C_1, C_2, \ldots, C_{K'}\}$. The definitions of precision, recall and F-Measure of the *i*-th Web service category C_i are given as below:

$$Precision(C_i) = \frac{|SC_i \cap C_i|}{|C_i|} \tag{9}$$

$$Recall(C_i) = \frac{|SC_i \cap C_i|}{|SC_i|} \tag{10}$$

$$F-Measure(C_i) = \frac{2*Precision(C_i)*Recall(C_i)}{Precision(C_i)+Recall(C_i)} \tag{11}$$

Here, the quantity of Web services in category SC_i is $|SC_i|$, the quantity of Web services in category C_i is $|C_i|$, the quantity of Web services jointly appeared in categories SC_i and C_i is $|SC_i \cap C_i|$.

Moreover, the purity and entropy of each C_i, and the purity and entropy of $EWSC$, are respectively defined as:

$$Purity(C_i) = \frac{1}{|C_i|} \max_j n_i^j \ , 1 \le i \le K', 1 \le j \le K \tag{12}$$

$$Purity(EWSC) = \sum_{i=1}^{K'} \frac{|C_i|}{|EWSC|} Purity(C_i) \tag{13}$$

$$Entropy(C_i) = -\frac{1}{logK} \sum_{j=1}^{K} \frac{n_i^j}{|C_i|} log(\frac{n_i^j}{|C_i|}) \tag{14}$$

$$Entropy(EWSC) = \sum_{i=1}^{K'} \frac{|C_i|}{|EWSC|} Entropy(C_i) \tag{15}$$

Where $|C_i|$ is the quantity of Web services in category C_i, n_i^j is the quantity of Web services within SC_j successfully split into C_i, and $|EWSC|$ is the overall quantity of Web services which must to be classified. In a word, if the recall, precision, and purity are higher and the entropy is lower, the accuracy of Web services classification is higher.

C. Baseline Methods

- TF-IDF+LR: exploiting the term frequency and inverse document frequency to measure the similarity among Web services. Logistic regression is used as the classifier of Web services.
- LDA: adopting the LDA topic model to classify Web services, and each Web service within the classified category has the highest topic possibility.
- WE-LDA [34]: the word vectors are used to increase the quality of Web service clustering. By utilizing K-means++ algorithm to cluster the word vectors acquired through Word2vec into word clusters. The word clusters are integrated to build a better representation for Web services.
- LSTM [35]: the input of LSTM is the eigenvector matrix of Web service descriptions, and the output is the prediction matrix of Web service classification. Only historical context (the pre-order information) can be utilized to classified Web services.
- Bi-LSTM [36]: there are two parallel LSTM layers in pre-order and post-order, which simultaneously utilize the historical contexts and the future context to classify the Web services.
- Wide & Deep [26]: it simultaneously exploits deep neural network learning and wide linear learning for model training, and combines the benefits of memorization and generalization to classify the Web services.
- Wide & Bi-LSTM [24]: it enhances the generalization capabilities of deep neural networks by replacing the deep components in the Wide & Deep model [26] with Bi-LSTM model [36], to achieve better quality of Web service classification.

D. Experimental Results and Analysis

The experimental comparisons are displayed in Figures. 4-8 while the quantity of Web service category varies from 10 to 50. Our experimental outcomes display that when the WSC-GCN model is applied to Web services classification, it has obvious advantages over other approaches. More concretely,

- Without tag information, the classification performances of other seven methods are all lower than that of the WSC-GCN. For instance, the precision of WSC-GCN without tag information individually has 85.3% increases to TF-IDF+LR, 70.6% increases to LDA, 30.2% increases to WE-LDA when the amount of service category equals to 50. This is because the network structure information embodied in words and Web service description documents can be fully exploited by the WSC-GCN model using convolution computation to gain a better classification quality.
- When the amount of Web service category is 40, the results of TF-IDF+LR, LDA and WE-LDA are the optimal in all situations. When the amount of category is undersized and increases from 10 to 40, the performance of Web service classification gradually moves up toward. This is because more latent information, such as topics, sematic correlations, can be learned and mined from these increasing Web services categories. Accordingly, the performance of Web service classification begins to drop with the continuous growing of the amount of category from 40 to 50. This is due to additional category only with fewer amounts of Web services and less content information. In addition, TF-IDF+LR have the worst performance in all situations.
- The precision of WSC-GCN without tag information obtains a rapid growth by 51.6% contrasted to the LSTM, and 19% contrasted to the Bi-LSTM. This is because the LSTM and Bi-LSTM only extract the context in the Web service description, but the network structure information contained in Web service description documents and words is ignored.

- Compared with the Wide&Deep and Wide&Bi-LSTM, the accuracy of WSC-GCN without tag information is improved by 36.5% and 5.5%. The reason is that the Wide&Deep and Wide&Bi-LSTM improve the classification effect of Web services through memory and generalization, but they also do not give consideration to the network structure contained in the Web service description documents and words.
- When the tag information is added, the precision of the WSC-GCN+tag model has 0.9%, 1.5%, 1.8%, 2.0%, 2.5% improvement respectively compared to the WSC-GCN model without the tag information (the amount of Web service category is 10/20/30/40/50, respectively). This shows that the addition of tag information enriches the corpus and semantics of the "Words & Web service description documents" heterogeneous graph network, and makes the Web service classification more accurate.
- As shown in the Figure.7, when the amount of Web service category is 50, the entropy of the WSC-GCN is the lowest, meaning that the classification result is the best contrasted to other models. In addition, we notice that when the amount of Web service category is 50, the purity of the WSC-GCN without tag information has 13.5% increases over Wide&Bi-LSTM.

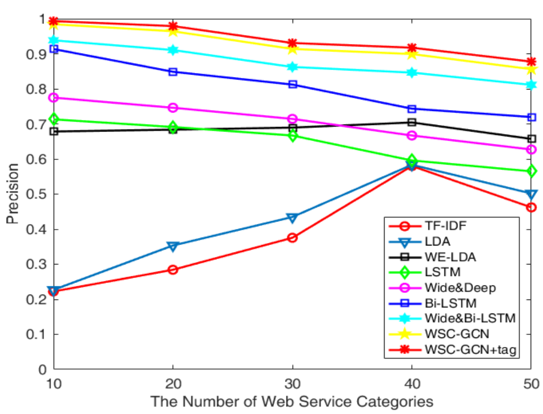

Figure 4. Precision

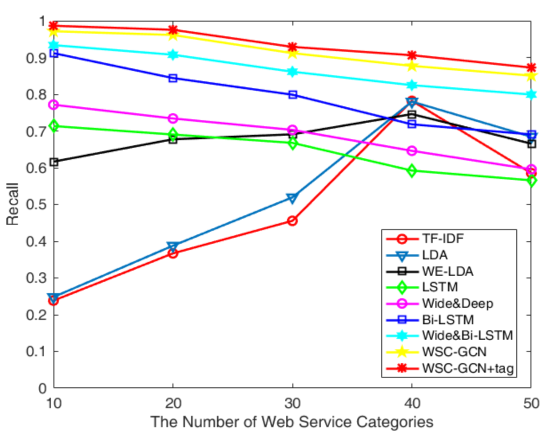

Figure 5. Recall

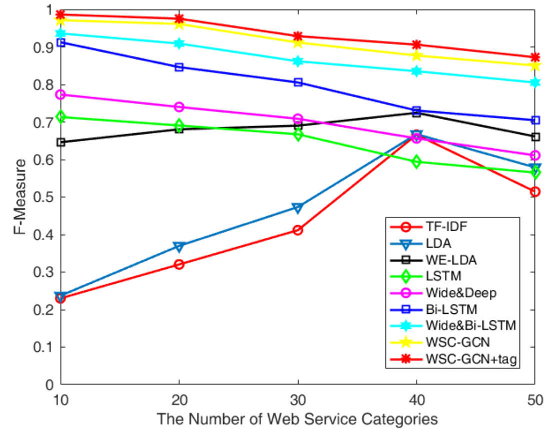

Figure 6. F-Measure

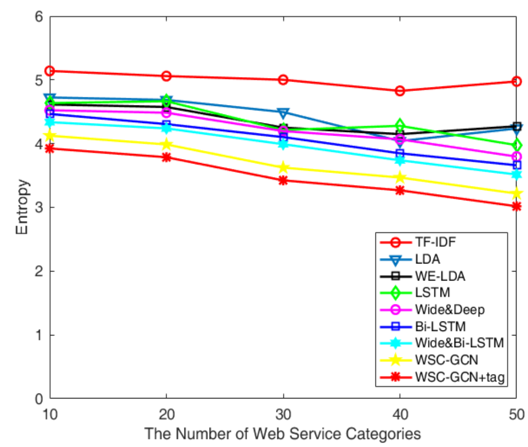

Figure 7. Entropy

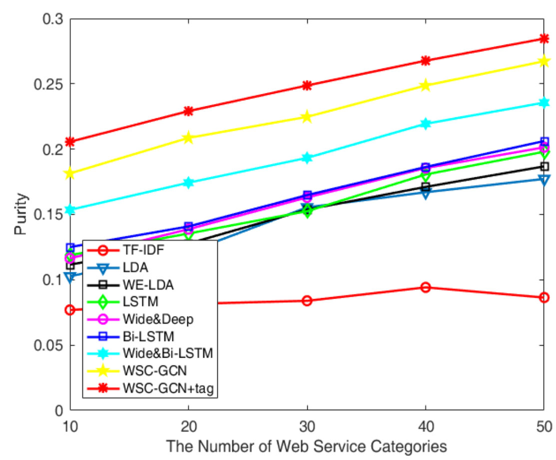

Figure 8. Purity

V. Conclusion and Future Work

A Web services classification approach on the basis of WSC-GCN model is presented in our paper. This method deeply mine the network structure information contained in the description document of Web services and its words. A "Words & Web service description documents" heterogeneous graph network for a whole Web service corpus is built to learn the embedding information of words and Web service description documents by using GCN, which transforms Web service classification into node classification problem. The experimental results on the dataset of ProgrammableWeb display that the proposed method is superior to the other methods in terms of the metrics of precision, recall, F-measure, purity and entropy. In the next work, we will further utilize the network representation information contained in service relationships or links to facilitate Web services classification.

Acknowledgement

Our work is supported by the National Natural Science Foundation of China (No. 61873316, 61872139, 61572187 and 61702181), the Educational Commission of Hunan Province of China (No.17C0642), and the Natural Science Foundation of Hunan Province (No. 2017JJ2098, 2018JJ3190 and 2018JJ2136). Buqing Cao is the corresponding author of this paper.

References

[1] L. Wang, W. Jie, J. Chen, Grid computing: Infrastructure, Service, and Applications, 2018.

[2] G. Zhang, Y. Yang, J. Chen, A historical probability based noise generation strategy for privacy protection in cloud computing, Journal of Computer and System Sciences 78 (5): 1374-1381, 2012.

[3] W. Li, Y. Yang, J. Chen, D. Yuan. A cost-effective mechanism for cloud data reliability management based on proactive replica checking, CCGrid 2012, pp. 564-571, 2012.

[4] C. Liu, R. Ranjan, X. Zhang, C. Yang, D. Georgakopoulos, J. Chen. Public auditing for big data storage in cloud computing--a survey, Proceedings of IEEE 16th International Conference on Computational Science and Engineering, pp. 1128-113, 2013.

[5] D. Puthal, S. Nepal, R. Ranjan, J. Chen, A dynamic key length based approach for real-time security verification of big sensing data stream, WISE 2015, 93-108, 2015.

[6] K. Wan, P. Lei, C. Chatwin and R. Yong, "Service-OrientedArchitecture". 2006, pp. 1-6.

[7] C. Roberto. "Web services description language (wsdl) version 2.0 part 1: Core language." W3C recommendation , 2007: 19.

[8] M. David. "Bringing semantics to web services: The OWL-S approach." in International Workshop on Semantic Web Services and Web Process Composition. Springer, Berlin, Heidelberg, 2004.

[9] X. Zhang, Y. Yin, M. Zhang, et al., "Web Service Community Discovery Based on Spectrum Clustering," in International Conference on Computational Intelligence and Security. IEEE Computer Society, Beijing, China, 2009, pp.187-191.

[10] J. Chen, Y. Yang, Temporal dependency based checkpoint selection for dynamic verification of fixed-time constraints in grid workflow systems, ICSE2008, 141-150, 2008.

[11] K. Elgazzar, A. E. Hassan, P. Martin, "Clustering WSDL Documents to Bootstrap the Discovery of Web Services", ICWS2018. IEEE Computer Society, 2010, pp.147-154.

[12] Q. Yu, H. Wang, L. Chen, et al., "Learning Sparse Functional Factors for Large scale Service Clustering," ICWS2015, New York, NY, USA, 2015, pp. 201 -208.

[13] L. Chen, Y. Wang, Q. Yu, Z. Zheng, J. Wu. "WT-LDA: User tagging augmented LDA for web service clustering". ICSOC2013, Springer, pp.162-176.

[14] H. Ye, B. Cao, Z. Peng, T. Chen, Y. Wen, J. Liu,. "Web Services Classification Based on Wide & Bi-LSTM Model. IEEE Access, 7, 2019: 43697-43706.

[15] M. Zhang, X. Liu, R. Zhang, et al., "A Web Service Recommendation Approach Based on QoS Prediction Using Fuzzy Clustering," SCC2012. IEEE, Honolulu, HI, USA, 2012, pp.138-145.

[16] H. T. Cheng, L. Koc, J. Harmsen, et al., "Wide & deep learning for recommender systems," in Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. ACM, Boston, MA, USA, 2016, pp. 7-10.

[17] H. Wang, Z. Yang, Q. Yu, "Online Reliability Prediction via Long Short-Term Memory for Service-Oriented System," ICWS2017, IEEE, Honolulu, Hawaii, USA, 2017, pp.81-88.

[18] M. Shi, J. Liu, D. Zhou, Y. Tang, "A topic-sensitive method for mashup tag recommendation utilizing multi-relational service data," IEEE Transactions on Services Computing, pp.99(1939):1-1, 2018.

[19] B. Cao, X. Liu, B. Li, et al., "Mashup Service Clustering Based on an Integration of Service Content and Network via Exploiting a Two-Level Topic Model," Presented at IEEE International Conference on Web Services (ICWS), San Francisco, CA, USA, 2016, pp. 212-219.

[20] B. Cao, X. Liu (Frank), J. Liu, M. Tang, "Domain-aware Mashup Service Clustering based on LDA Topic Model from Multiple Data Sources," Information and Software Technology, vol.90, 2017, pp.40-54.

[21] M. Shi, J. Liu, D. Zhou, et al., "WE-LDA: A Word Embeddings Augmented LDA Model for Web Services Clustering," in IEEE International Conference on Web Services (ICWS), Honolulu, HI, USA, 2017, pp.9-6.

[22] M. Shi, Y. Tang. J. Liu, "Functional and Contextual Attention-based LSTM for Service Recommendation in Mashup Creation," IEEE Transactions on Parallel and Distributed Systems, pp:1-1, 2018.

[23] S. Li, Z. Yan, X. Wu, et al., "A Method of Emotional Analysis of Movie Based on Convolution Neural Network and Bi-directional LSTM RNN," in IEEE Second International Conference on Data Science in Cyberspace. IEEE, Shenzhen, China, 2017, pp.156-161.

[24] K. Thomas, and M. Welling, "Semi-supervised classification with graph convolutional networks." arXiv preprint arXiv:1609.02907 ,2016.

[25] B. Cao, X. Liu, M. Rahman, et al., "Integrated Content and Network-Based Service Clustering and Web APIs Recommendation for Mashup Development," IEEE Transactions on Services Computing, vol. 1, no.1, 2017.

[26] L. Yao, C. Mao, Y. Luo. "Graph convolutional networks for text classification." arXiv preprint arXiv:1809.05679 , 2018.