Web Service Classification Using Support Vector Machine

Conference Paper · November 2010

DOI: 10.1109/ICTAI.2010.9 · Source: IEEE Xplore

6 authors, including:

Xuan Zhou

70 PUBLICATIONS **639** CITATIONS

SEE PROFILE

Athman Bouguettaya

University of Sydney

298 PUBLICATIONS **4,331** CITATIONS

SEE PROFILE

2010 22nd International Conference on Tools with Artificial Intelligence

Web Service Classification using Support Vector Machine

Hongbing Wang*, Yanqi Shi*, Xuan Zhou†, Qianzhao Zhou*,
Shizhi Shao* and Athman Bouguettaya†
*School of Computer Science and Engineering,
Southeast University, China
Email: {hbw,yqs,qzz,szs}@seu.edu.cn
†CSIRO ICT Centre, Australia
Email: {xuan.zhou, athman.Bouguettaya}@csiro.au

***Abstract*—Classification is a widely used mechanism for facilitating Web service discovery. Existing methods for automatic Web service classification only consider the case where the category set is small. When the category set is big, the conventional classification methods usually require a large sample collection, which is hardly available in real world settings. This paper presents a novel method to conduct service classification with a medium or big category set. It uses the descriptive information of categories in a large-scale taxonomy as sample data, so as to disengage from the dependence on sample service documents. A new feature selection method is introduced to enable efficient classification using this new type of sample data. We demonstrate the effectiveness of our classification method through extensive experiments.**

I. Introduction

With the increasing number of available Web services on the internet, Web service discovery becomes a challenging issue. It is time consuming to traverse an entire Web service collection to find a matching service. To speed up service discovery, classification can be applied. The existing work on Web service classification has mainly focused on text classification and property similarity computing methods. In [1], the textual description in a WSDL document is mapped into a feature vector and then automatically classified into domain-specific categories using SVM method, but it ignores the semantic features of the service document and the classification accuracy is low. In [2], it uses binary feature vector and judges the occurrence of term only if the document directly contains this term or contains some terms are equivalent with this term in ontology. This will lead to information loss when semantically related concepts occur in the document. In [3], the similarity between two services is measured in terms of the similarity of their operations and parameters. However, the measure for comparing the complete set of parameters is complex and domain dependent. This increases the difficulty of implementation. To the best of our knowledge, the existing methods only considered the case when the category set is very small, while a practical classification system in the real word setting usually needs to deal with thousands of categories.

This paper presents a hierarchical classification system to classify Web services based on their functional features. The system utilizes the taxonomy of UNSPSC[1] of which the category set is very large. We use the descriptive information of each category as the sample documents of its parent category, so that our system does not rely on a collection of pre-classified services. A new feature selection method is proposed to map a high dimension feature space to a low dimension space based on semantic similarity of concepts.

[1]United Nations Standard Products and Service Code. http://www.unspsc.org

II. Overview of the Classification System

In this section, we introduce the taxonomy and classification algorithm of our classification system, and give a brief overview of the classification process.

A. The UNSPSC Taxonomy

It is desirable that the classification of services be based on a standard and widely used taxonomy, so that users could easily find the requested services in the expected categories. UNSPSC is one of the standard taxonomies established for E-commerce products and services. The structure of UNSPSC is a five-level and tree-structured hierarchical classification. The five levels are root node, segment, family, class and commodity respectively. In this paper, we use UNSPSC as the classification criteria for Web services. As the categories in commodity level are concrete products or services, we only categorize services into the class level of the UNSPSC.

B. The SVM Classification Algorithm

We use the SVM(Support Vector Machine) text classification algorithm to classify the service documents. The SVM method was first introduced to conduct text classification by Joachims[4]. The process of text classification using SVM method can be summarized as follows: First, we get a document collection $\Omega = \{d_1, \ldots, d_{|\Omega|}\} \subset D$ such that each document d_i corresponds to a category c_j in $C = \{c_1, \ldots, c_{|C|}\}$ and a feature space $T = \{t_1, t_2, \ldots t_{|T|}\}$ which contains all the different terms in Ω. Then, each sample document is mapped to a feature vector $W = \{w_1, w_2, \ldots w_{|T|}\}$, where w_i denotes the weight of the term t_i in this document. (If w_i=0, it denotes that term t_i does not appear in this document.) Following that, all the feature vectors are input to the SVM to train the classifier. Finally, the feature vector of an unclassified document is passed to the SVM classifier to find the category of the document.

1082-3409/10 $26.00 © 2010 IEEE
DOI 10.1109/ICTAI.2010.9

C. The Process of the Service Classification

In the process of service classification, a classifier need to be trained previously, e.g. using the SVM mechod. When registering a new Web service, the functional descriptions, such as the input and output information, will be extracted from the service document to map to a feature vector. Then the feature vector is input into the classifier to determine which category the service belongs to. In order to reduce the complexity of the problem, we will categorize Web service from top to bottom. After obtaining the category, users will confirm whether the service is classified into the proper category, if not, users can classify the service manually. At last, the service is added to the service list of the category. Figure 1 shows the complete process of service classification in our system. The following section will give a detailed introduction about the classification of the Web service.

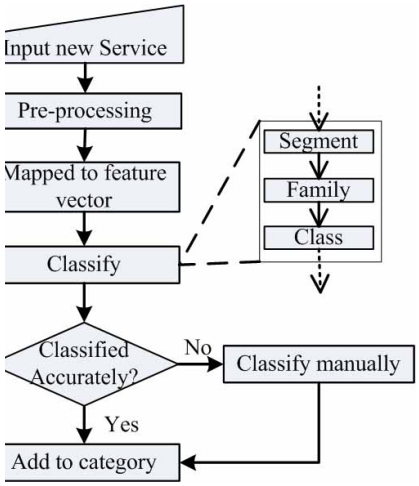


Fig. 1. The complete process of service classification

III. Classification in the UNSPSC Taxonomy

Services will be classified into the categories of the Class layer in UNSPSC. As mentioned in the introduction, there is no comprehensive collection of service documents can serve as the sample collection of the category set in UNSPSC. To solve this problem, in this paper, we propose a new and practical training scheme. Our classification system mainly focuses on the functional features of Web services rather than the whole service document. This will not only improve the accuracy of classification but also broaden the application scope of our method, as we can extract the functional description of services from any document formats.

A. Hierarchical Classification

Due to the large number of categories in UNSPSC, if we train the classifier to distinguish all the bottom categories directly, the dimension of the feature space will be very high. We use hierarchical classification method to make the classification limited to only a branch of the taxonomy tree. During training the classifier, we only need to distinguish the sibling nodes under the same parent category, rather than to distinguish all categories in the same level of UNSPSC. In this way, the scale of the feature space of each sub-classification will be reduced. As a result, the efficiency of the classification system can be improved significantly[5]. We treat the UNSPSC as a Multi-level tree, where each of non-leaf nodes (parent class) corresponds to a sub-classification system. In each sub-classification system, the definitions of the sub-categories are treated as the sample documents of the parent category.

B. Feature Selection

We assume that the category collection is $C = \{c_1, c_2, \ldots c_p\}$. Under each category c_i, the definitions of all the direct sub-categories $d_{i1}, d_{i2}, \ldots$ can be treated as sample documents of c_i. After removing the stop words in the sample document, we get the feature space $T = \{t_1, t_2, \ldots, t_n\}$ corresponding to the category collection C. Then each document d_e is mapped into a feature vector $W_e = \{w_{e1}, w_{e2}, \ldots, w_{en}\}$. The functional description of a Web service is always related to a set of concepts. However, as the concepts are defined in different abstraction levels, the concepts of the feature space and concepts of service document may not match. For example, there are category set C={Transportation, Financial} and a new service with the description "Scheduled bus service". If the feature space T={Cargo, Passenger, Auditing, Insurance}, then the service will be mapped into the feature vector $W = \{0, 0, 0\}$. Actually, the feature "passenger" contains the feature "bus" in its sub-categories. Therefore if the concepts in the feature space are too abstract, it will result in lower classification accuracy.

In order to improve the accuracy of classification, we generate feature vectors by the concepts in the Commodity layer belongs to category set C, supposing it is $T_{com} = \{tt_1, tt_2, \ldots tt_m\}$, and treat the definition of each Commodity category under C as a sample document. On the other hand, although SVM can scale up to considerable dimensionalities, some researches[6][7] have demonstrated that none of the feature selection methods tested improved SVM classification accuracy in higher dimensions. Hence, we will establish a mapping between T_{com} and T, reducing the dimension of the term space to improve the classification efficiency. The mapping function is defined as follows:

$$f(tt_i) = \{t_j \in T | \forall t_k \in T, SemSim(tt_i, t_j) \geq SemSim(tt_i, t_k)\}$$
$$i \in (1, 2, \ldots m), j \in (1, 2, \ldots n)$$

$SemSim(tt_i, t_j)$denotes the semantic similarity of tt_i and t_j calculated using WordNet[2].

WordNet only provides semantic similarity of concepts when they are both nouns or are both verbs. To measure the similarity between adjectives or adverbs, we change derivative adjectives or adverbs to their corresponding nouns or verbs. For example, the adjectives with the suffix "-ic" are mostly

[2]http://wordnet.princeton.edu/

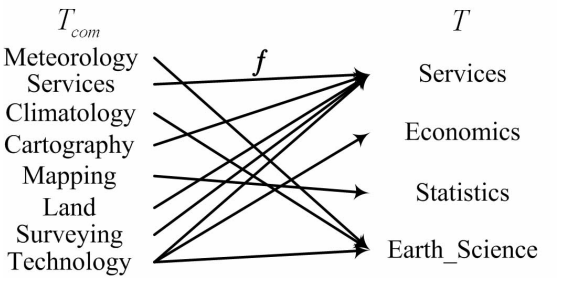


Fig. 2. The mapping between T_{com} and T

evolved from nouns and the adjectives with the suffix "-able" are mostly evolved from verbs. After obtaining the feature space, the definition of each category in Commodity level belongs to C will be mapped into a feature vector $W_e = \{w_{e1}, w_{e2}, \ldots, w_{en}\}$, where w_{ei} is the cumulative average of semantic similarity between t_i and all the concepts having a mapping with t_i. The calculating formula is as follows:

$$w_{ei} = \frac{\sum_{s}^{f(tt_s)=t_i} SemSim(t_i, tt_s)}{|\{tt_s\}|}$$

In this formula, $|\{tt_s\}|$ denotes the number of concepts in T_{com} mapped to t_i. We do not consider term frequency, because the description of a sample document is general and brief, and the term frequency of a concept is not correlated with its importance.

The following example illustrates the generation process of a feature vector. Assume that T ={*Services, Economics, Statistics, Earth_Science*}, T_{com} ={*Meteorology, Services, Climatology, Cartography, Mapping, Land, Surveying, Technology*}, and the sample document is "Meteorological services". First, we establish the mapping f between T and T_{com}, and use the Path Length[8] method to calculate the semantic similarity between concepts in the WordNet. The mapping is shown in figure 2. We can see that each feature(term) in T_{com} is mapped to one or more features in T. The terms "meteorological" and "services" corresponds to the features "Meteorology" and "Services" in T respectively, "Meteorology" is mapped to the feature "Earth_Science" in T_{com} with the similarity 0.5, and "Services" is mapped to the feature "Services" with the similarity 1. Then the sample document can be mapped into a feature vector W={1, 0, 0, 0.5} according to formula 1.

When all the feature vectors of the sample documents are generated, the classifier can be trained. For each category(node) in the root, Segment and Family level in UNSPSC, there will be a classifier to classify a service into its subcategories.

C. Classification of New Service

When all the classifiers are obtained, the text contents about the functionality of a new service S_c can be mapped into a feature vector $W_c = \{w_{c1}, w_{c2}, \ldots, w_{cn}\}$ too. The calculation of w_{ci} is the same as that of w_{ei}. After the textual description about a service's functionality is mapped to a feature vector, the feature vector is fed to our classifier to determine its category.

TABLE I
EFFECTIVENESS OF THE CLASSIFICATION SYSTEM

Similarity Measure	Micro-F_1		Micro-BEP	
	Poly	RBF	Poly	RBF
Path Length	89.0%	90.5%	89.1%	90.5%
JCn	85.1%	86.0%	85.3%	86.2%
Wu&Palmer	86.5%	87.4%	86.5%	87.5%

IV. EXPERIMENT

We used the OWLS-TC[3] Version 3.0 as our test collection. It consists of 1007 Web services described using OWL-S. The service descriptions are pre-classified in seven categories, namely *Travel, Education, Weapon, Food, Economy, Communication,* and *Medical*. The contents in the label of *serviceName, textDescription, hasInput, hasOutput, hasPrecondition* and *hasResult* were extracted as textual descriptions of the services. There are several measures to calculate the semantic similarity between two words in the WordNet. We selected three representative ones: Path Length, Wu & Palmer[9] and JCn[10]. We used a large number of simulated categories or services to test the scalability of our classification system. The sample documents of the simulated categories were generated by composing pieces of texts of several categories under the same commodity category of UNSPSC. The feature vectors of the simulated services were directly generated from the feature space of the service documents of OWLS-TC.

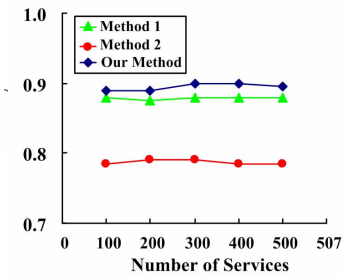
A. Efficiency and Effectiveness of the Classification System

First, we conducted several experiments on the entire OWLS-TC test collection to study the evolution of the performance measures. Table 1 reports the micro-averaging F_1 and BEP when classifying with three different semantic similarity measures and two different SVM kernel functions. As we can see, the semantic similarity measure Path Length achieved the best results, as it is both stable and effective, and the JCn measure performed the worst. This can be attributed to the difference of the value range of the similarity measure. The value range of the words similarity of JCn is [0,+∞] while the other two are [0,1]. The overly large similarity values in the feature vector would obscure the effects of other features whose similarity value is small. On the other hand, the performance of the SVM classifier with the RBF kernel function is better than the one with polynomial kernel function.

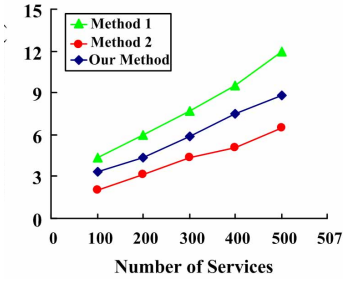
In our next set of experiments, we compared our results using the Path Length similarity measure and the SVM classifier with RBF kernel function with other service classification methods. To make the results comparable, we applied these methods both on the test collection OWLS-TC. 500 services

[3]http://projects.semwebcentral.org/projects/owls-tc/

were selected as these two methods' common sample collection, and the other 500 as the test collection. We used two existing classification methods as alternatives. Method 1[2] uses SVM as the classifier and combines the textual and semantic information of the service documents as classification features. Method 2[3] uses property value comparison methods. Figure 3 shows the result. As we can see, our method offers the best effectiveness. As to time performance, three method performed equally well.



(a)



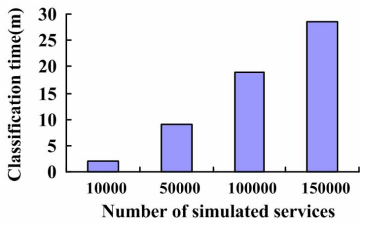
(b)

Fig. 3. Comparation with other classification methods

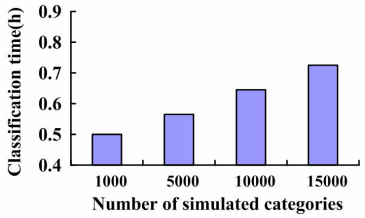
Finally, we assessed the scalability of our classification system. We varied the number of simulated categories from 1000 to 15000 (there were 100000 simulated services in total), and the number of simulated services from 10000 to 150000 (assuming 10000 categories). The results are shown in Figure 4. As shown in the results, the time performance is acceptable even when the number of services are very big. When the number of services is fixed, the classification time grows slowly with the increase of the number of categories, that is, the classification time of a SVM classifier is proportional to the number of categories it has to distinguish.

V. Conclusion

In this paper we present a function-oriented Web service classification system which uses taxonomy of UNSPSC as the category set. By utilizing the descriptive information of the sub-categories as the sample data, we solve the problem of inadequate sample service documents to classify new services into large-scale taxonomy like UNSPSC. A new feature selection method by mapping a high dimension feature space into a lower one using semantic similarity of concepts is proposed which can reduce the dimension of feature space, while reducing the loss of the document features. The experimental results



(a)



(b)

Fig. 4. Scalability of our classification method

validate the efficiency and effectiveness of our classification method.

References

[1] Marcello Bruno, Gerardo Canfora, Massimiliano Di Penta, and Rita Scognamiglio. An approach to support web service classification and annotation. In *Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05)*, pages 138–143, Washington, DC, USA, 2005.

[2] Ioannis Katakis, Georgios Meditskos, Grigorios Tsoumakas, Nick Bassiliades, and Ioannis P. Vlahavas. On the combination of textual and semantic descriptions for automated semantic web service classification. In *AIAI*, volume 296 of *IFIP*, pages 95–104. Springer, 2009.

[3] Miguel Angel Corella and Pablo Castells. P.: A heuristic approach to semantic web services classification. In *Proceedings of the 10th International Conference on Knowledge-Based & Intelligent Information & Engineering Systems (KES 2006*, 2006.

[4] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Machine Learning: ECML-98*, pages 137–142. Springer, 1998.

[5] Michelangelo Ceci and Donato Malerba. Classifying web documents in a hierarchy of categories: a comprehensive study. *J. Intell. Inf. Syst.*, 28(1):37–78, 2007.

[6] Minh Hoai Nguyen and Fernando de la Torre. Optimal feature selection for support vector machines. *Pattern Recogn.*, 43(3):584–591, 2010.

[7] Johan Bjorkegren Roland Nilsson, Jose M. Pena and Jesper Tegner. Evaluating feature selection for svms in high dimensions. In *17th European Conference on Machine Learning Berlin*, pages 719–726, Berlin, 2006. Springer.

[8] Roy Rada, Hafedh Mili, Ellen Bicknell, and Maria Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems Management and Cybernetics*, 19(1):17–30, 1989.

[9] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138, Morristown, NJ, USA, 1994. Association for Computational Linguistics.

[10] J.J. Jiang and D.W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. of the Int'l. Conf. on Research in Computational Linguistics*, pages 19–33, 1997.