A Kernel based Structure Matching for Web Services Search

Jianjun Yu[*] Shengmin Guo Hao Su Hui Zhang Ke Xu
State Key Lab. of Software Development Environment
Beihang University, Beijing, China
{yujj,guosm,suhao,hzhang}@nlsde.buaa.edu.cn

ABSTRACT

This paper describes a kernel based Web Services (abbreviated as service) matching mechanism for service discovery and integration. The matching mechanism tries to exploit the latent semantics by the structure of services. Using textual similarity and n-spectrum kernel values as features of low-level and mid-level, we build up a model to estimate the functional similarity between services, whose parameters are learned by a Ranking-SVM. The experiment results showed that several metrics for the retrieval of services have been improved by our approach.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*retrieval models, search process, selection process*; H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Web-based Services*

General Terms

Algorithms, Languages, Design, Experimentation

Keywords

Web Services, Web Services Matching, WSDL, n-Spectrum Kernel, Ranking SVM

1. INTRODUCTION

A hotspot in the research of service is to realize large-scale service discovery and integration in Internet. Conventional matching mechanisms, like in UDDI, are mainly based on keyword search, however, the precision of those approaches are relatively low. To address this problem, several approaches considering structural information have sprung up [1, 3, 6]. They significantly improve the precision with the requirement that matched services should be similar in structure. But in practical, those approaches are too strict to recognize similar services with different data structure encapsulation. Consequently, they have low recalls.

Therefore, we put forward a novel loose tree matching algorithm which extracts the structural features from another perspective to help improve the recognition of functionally similar services.

2. WEB SERVICES SIMILARITY

In our algorithm, services are schemed by WSDL (Web Services Description Language) as tree-structured documents, and two kinds of features are extracted.

One kind is to calculate two WSDL documents' text similarity with a classical VSM (Vector Space Model) [2], regarding the documents as unstructured text. In the classical VSM, a document is formalized as a vector, and each dimension of the vector represents a word in the document. The value of the dimension is calculated by the $tf - idf$ formula [2] as an estimation of the importance of the word in the document.

The other kind of features describes the structural similarity. It takes two steps to get such features. First, we have to do some preprocess to extract document trees from the two WSDL documents and align their nodes according to the label's textual similarity. Aligned nodes are considered identical. After the preprocess, we model the documents trees as a vector in a n-spectrum vector space ($n = 2, 3, ...$), and use the newly brought forward n-spectrum kernel function [4] to compare how much hierarchical relationships the two trees share in common. With a set of different n, we have a set of different function values as features.

Modeling a document tree as a vector in the n-spectrum vector space is similar to modeling a text document stated above, except that each dimension in the n-spectrum vector space represents an n-path subsequence defined as below:

A *path subsequence* (abbreviated as ps) is an ordered set of nodes extracted from a path of the document tree (abbreviated as DT). The set containing all $n - ps$ (n is the number of nodes in a ps) of the DT is called the n-spectrum of DT, denoted by $\Psi^n(DT)$. Intuitively, two document trees are more similar if they share more common ps.

Then, we consider the problem of a $ps's$ weight in the vector. A ps may not be picked contiguously from a path, however, we assume that contiguous picked ps should weigh more. So, we define a decay factor $\lambda \in (0, 1)$ to weight the presence of gap in a ps. For the indices p identifying the occurrence of a ps in a document tree DT, we use $l(p)$ to denote the distance between the first node and the last node in the corresponding path, namely, the difference of their depth in the tree. In the gap-weighted kernel, we weight the occurrence of the ps with the exponentially decaying weight $\lambda^{l(p)+1}$.

With the set of $n - ps$ and their weights in the vector, we can give the formula of n-spectrum kernel value:

DEFINITION 1 (GAP-WEIGHTED N-SPECTRUM). The feature space F associated with the gap-weighted spectrum

[*] Supported by 973 2005CB321901 and BNP 2005B12

WWW 2007, May 8–12, 2007, Banff, Alberta, Canada.
ACM 978-1-59593-654-7/07/0005.

kernel of n-ps is indexed by $I = \Sigma^n$ (the set of all n-ps), with the embedding $\phi^n : DT \mapsto (\phi^n_{ps}(DT))_{ps \in I} \in F$ given by

$$\phi^n_{ps}(DT) = \sum_{p:ps=DT(p)} \lambda^{l(p)+1}, ps \in \sum{}^n . \quad (1)$$

Where p is the indices identifying the occurrence of ps. The associated kernel is defined as:

$$\kappa_n(s,t) = \frac{< \phi^n(s), \phi^n(t) >}{\parallel \phi^n(s) \parallel * \parallel \phi^n(t) \parallel} = \frac{\sum\limits_{ps_s \Leftrightarrow ps_t} \phi^n_{ps_s}(s)\phi^n_{ps_t}(t)}{\parallel \phi^n(s) \parallel * \parallel \phi^n(t) \parallel} \quad (2)$$

where $ps_s \in \Psi^n_s$ and $ps_t \in \Psi^n_t$.

With the textual and structural similarities, we can estimate the functional similarity as follows:

$$Sim_k(ws_1, ws_2) = \alpha_1 sim_t(ws_1, ws_2) + \sum_{i=2}^{k} \alpha_i \kappa_i(DT_1, DT_2) \quad (3)$$

where $k \geq 2$, ws_1 and ws_2 are two services, DT_1 and DT_2 are their corresponding document trees, $sim_t(ws_1, ws_2)$ is their textual similarity, and κ_i is the gapped i-spectrum kernel describing their structural similarity.

The parameters α_i are estimated by the Ranking SVM [7]. Ranking SVM is brought forward by machine learning researchers in recent years to solve ordinal regression problem, and it has been proved to be effective [7]. In this paper, the retrieval of matched services is formalized as a "learning to rank" problem, where the matching degree between query and returned services are mapped into three ordered categories (ranks). We have three features in the ranking problem: text similarity, 2-spectrum and 3-spectrum kernel values, and their weights are in Equation (3). Our experiment result shows that the best values of λ are $\lambda_2 = 0.58, \lambda_3 = 0.85$ and the best values of α are $\alpha_1 = 0.55, \alpha_2 = 0.44, \alpha_3 = 0.01$ in our data sets.

3. EXPERIMENTS AND EVALUATION

The experiment data set of services is collected from WWW, containing 2,140 valid services in total.

In this paper, we define three categories for the judgment of the matching degree between the query and the WSDL documents: 'good, 'insignificant' and 'bad'.

Generally, we hope that a good return can be modified with little effort to replace the query in Web environment. Therefore, it should provide almost the same function to the query. A bad return may only be textually similar to the query document with no obvious structural similarity. An insignificant return is the one with a similarity degree between good and bad.

As a baseline method, we simply use the text similarity to rank and categorize the candidates into three categories. This method does not take structural information into consideration at all. As another baseline method, we use the Edit Distance from Gabriel Valientes book [5] to get the similarity between WSDL documents.

We made use of four evaluation metrics, which are "error rate", "R-precision", "Top-N precision", and "Average precision". The evaluation result on the test set is listed in Figure 1.

We have our evaluation on two levels. The first group is evaluated using 'good', while the second group is evaluated by replacing 'good' with 'good' and 'insignificant'. Taking a closer look at Figure 1, we notice that Ranking-SVM outperforms other approaches in every metric. As an interpretation of the result, we may regard VSM as low-level feature and 2-spectrum and 3-spectrum as mid-level features with structural information. The low-level feature is directly extracted from WSDL with less deeper semantic and less noise, while mid-level features reflect deeper semantic with more noises. Ranking-SVM combines VSM, 2-spectrum kernel and 3-spectrum features together, so that low-level feature and high-level features complement each other; as a result, it outperforms all other methods. We may also notice that the R-precision and Top-5 precision for 'good' are relatively equal. In fact, the average number of relevant documents to the queries is approximately 5.

Quality	Approach	Error-rate	R-precision	Top5 precision	Top10 precision	AP
Good	VSM	0.1850719	0.6705897	0.6666667	0.6266667	0.7235433
	ED	0.4079808	0.4665233	0.6266667	0.4733333	0.6436735
	2-spectrum	0.2000336	0.6910073	0.7066667	0.6466666	0.7636632
	3-spectrum	0.1976658	0.6591749	0.6800001	0.6200000	0.7327877
	R-SVM	**0.1740201**	**0.7583206**	**0.7866668**	**0.6666666**	**0.8098783**
Good & Insignificant	VSM	0.1825591	0.6892766	0.8400000	0.8133333	0.7845165
	ED	0.4079808	0.5607469	0.7866667	0.6266667	0.6870136
	2-spectrum	0.2000336	0.7456499	0.8533334	0.8266667	0.8153685
	3-spectrum	0.1976658	0.7383915	0.8400000	0.8200000	0.8158891
	R-SVM	**0.1740201**	**0.7615644**	**0.8933334**	**0.8466666**	**0.8312675**

Figure 1: The Evaluation Results. ED is the Abbreviation of Edit Distance.

4. CONCLUSION

In this paper, we present a novel approach for services matching problem. In order to achieve the task, we model WSDL documents in the vector space and then defined the gapped n-spectrum kernel function in the space. Using textual similarity and n-spectrum kernel values as features of low-level and mid-level, we build up a model to estimate the functional similarity between services, whose parameters are learned by a Ranking-SVM. Experimental results indicate that our model significantly outperforms other methods. Since the n-spectrum kernel function is defined in vector space, the framework of our approach can be easily adapted to matching problems of other domains.

5. REFERENCES

[1] D. Carmel, Y. S. Maarek, M. Mandelbrod, Y. Mass, and A. Soffer. Searching XML Documents via XML Fragments. In the Proceedings SIGIR'03, 2003.

[2] B. Everitt. Cluster Analysis. 2nd edition. *New York:Halsted Press*, 1980.

[3] N. Kokash. A Comparison of Web Service Interface Similarity Measures. University of Trento. Technical Report:DIT-06-025, 2006.

[4] J. Shawe-Taylor and N. Cristianini. Kernel Methods for Pattern Analysis. *United Kingdom: Cambridge University Press*, 2004. pp.344-372.

[5] G. Valiente. Algorithms on Trees and Graphs. *Springer-Verlag, New York*, 2002.

[6] Y. Wang, S. Eleni, O. M. E., W. Sanjiva, P. M. P., and J. Yang. Semantic Structure Matching for Assessing Web Service Similarity. In the Proceeding of ICSOC'03, 2003.

[7] J. Xu, Y. Cao, H. Li, and M. Zhao. Ranking Definitions with Supervised Learning Methods. In the Proceedings of WWW, 2005.