Middle-East Journal of Scientific Research 24 (7): 2239-2246, 2016
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2016.24.07.23695

Bridging the Semantic Gap in Web Services Discovery Through Support Vector Machines and Feedback Mechanism

[1]K. Venkatachalam and [2]N.K. Karthikeyan

[1]Dept. of Information Technology, Sri Krishna College of Engineering and Technology, Coimbatore, India
[2]Department of IT, Karpagam College of Engineering, Coimbatore, India

Abstract: The number of Internet users worldwide has increased from 1.02 billion in 2005 to 3.17 billion in 2015. Most time, these users keep searching for the data they require. With the advancement in digital communication and data sets getting huge over the internet, the process of finding an appropriate web service for a given task also increases. There is always a gap between the user specifications given in Natural Language (NL) and the web service definition which is given in a standard interface method like Web Services Description Language (WSDL). Traditional classification methods used in web services discovery fails when the category sets increases. So, we propose to use Support Vector Machines (SVM) for bridging this semantic gap along with feedback mechanism in retrieving the relevant web services. One of the popular measures of performance, precision and recall is used and we find that recall-precision break-even point can be reached with this feedback mechanism.

Key words: Natural language · Support Vector Machines · Clustering · Web Services Description Language · Break-even point · Web Ontology Language

INTRODUCTION

Web server contains discovery documents and when a URL is given to that, the client application developer can understand the existence of the web service along with its capabilities and the interaction mechanism which is known as said to be Web service discovery. Web service provider will normally publish a service that will be used by the web service consumer. We cannot expect both of them to use a common language and this is the main challenge in service-retrieval process. Present solutions are either Linguistic based or Semantics based. Linguistic methods do textual analysis of web service descriptions but lag in supporting a fully automated solution. The semantic approach includes Web Ontology Language (OWL) which is a prescribed way to define taxonomies and classification networks for different domains. But this methods lags in feasibility by needing the availability of full sematic models of the service query. This motivates the need to develop a system that has the ability to learn by itself without explicitly programmed.

The tools for web services discovery will find the URLs of XML web services which are found on the web server and it saves the documents which are related to each of these services on a local storage. The biggest challenge here is on identifying the service a client is interested among a huge collection of web services and various providers. Though the client is not interested in knowing the details of the provider who is providing the service, the quality of output is expected to be good.

Service registry is needed for web services discovery. Web service provider must provide appropriate description like trade, service, technical details etc. for publication of a service. These details are stored in the registry which can be document based or metadata based registry. Web service discovery can now locate service providers and retrieve web services descriptions that are published as shown in Fig 1. The architecture of this service registry plays a major role in web service discovery. The client application now gets the location, web service capabilities and interface mechanisms to proceed further. For facilitating this web service discovery, classification is a widely used mechanism and in our work we use SVMalong with feedback mechanism for improved results.

Corresponding Author: K. Venkatachalam, Dept of Information Technology, Sri Krishna College of Engineering and Technology, Coimbatore, India.

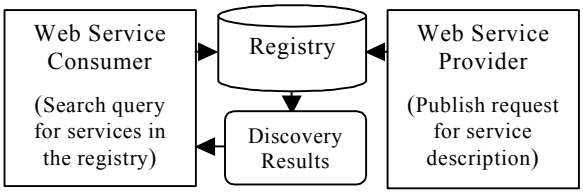

Fig. 1: Web Services Discovery Block Diagram

Relevant Work: For a common man to relate with the semantic web content, several steps were taken including Natural Language interfaces to ontologies and semantic information bases. Hongbing Wang et-al has proposed a method [1] where service discovery is introduced in the early Requirement Engineering stages to guide the decomposition process. Pablo Rodriguez-Mier and team has presented a graph-based framework [2] concentrated on the semantic input-output factor matching of service interfaces that ef?ciently incorporates the instinctive service composition and semantic web service finding. Zhang [3] says web services are growing technologies for constructing service oriented distributed systems and this discovery gets high importance for web service related applications.

Funda has described a lot about semantic web services' discovery in his master degree thesis [4] where he mentions that users find it difficult to identify services as the number of web services increases and it is also difficult for them to filter and add these services based on their requirements. He believes that the schema matching algorithms will identify the mappings between different schema models and helps improve discovery.

Cermi in his book [5] details about writing the own services and it helps programmers the basics of web services along with references to XML web services by demonstrating the easiest ways to create services by using Java tools.Kreger in his report on web services conceptual architecture [6] mentions that the publication of web services includes producing service descriptors along with publishing. He shows the range from the most stagnant, easiest technologies for web service publish and discovery to the most active, more complex technologies.

Venkatachalam *et al.* [7] reviews the web service discovery mechanism along with the composition concepts and also analyzes the same from different perspectives and has identified probable futuristic research areas in this topic of interest in their survey paper.

Eran Toch and team [8] have investigated different aspects of web services retrieval by bridging the gap between user query and the web service definitions written in WSDL. They have though limited to do additional classification work and tried to make the services accessible without classification.

Pavithra G *et al.* [9] has surveyed and explained a technique for searching the information from the repository where the keyword is used with incremental construction of query for better search and retrieval mechanisms.Bouziane along with his team has brought out the importance of question answering systems along with the required statistics and analysis [10]. They have proposed a new system for complex queries and showed improvement over the existing techniques.

Hongbing Wang *et al.* [11] has discussed about classification in their research paper. They feel the current methods are useful only for small category set and needs more improved mechanisms for larger sets. They have used support vector machines based classification using a big category set.

Zi Long Chen and Yang Lu [12] feel the feedback mechanisms could improve the performance of retrieval. They train the classifier with the help of feedback documents. The system is then used for classifying the test documents.

Design: The question answer system retrieves the user's question in the pertinent way as per the requirement. We have designed a system with the support vector machine based classification which tries to provide better results as compared to the traditional methods discussed in the previous section.

The basic requirement here is to design a system that can learn by itself in classifying the user query and come up with the relevant results with better accuracy than the existing methods. There are a variety of machine learning algorithms that can help in this case and one such machine learning algorithm is Support Vector Machine which is supervised learning models and can be used effectively in classification. They are trained initially with few examples and the system builds a model based on the training finally resulting in classifying the test vectors in to one of the two categories, making it a non-probabilistic binary linear classifier. Feedback techniques in general help the retrieval performance but the methods used sometimes yield totally unrelated results thereby reducing the retrieval accuracy. So, we design the feedback mechanism in a better way which can classify the web services after initial training by itself which in turn helps improve the precision-recall parameters.

Ontology is another important factor that needs to be considered in semantic web searching mechanisms. It actually helps in better communication with the system. It

refers to the definition of the types, possessions and interrelationships of the objects that belong to a specific domain of discourse. The three basic components include classes, attributes and relationships. Fig 2 represents one such domain which is ontology for a country. It includes several objects like location, capital, leaders, famous places, languages, religion, area, currency etc. They enable knowledge sharing and exchange. To get the information from database, knowledge base in ontology provides an improved way.

We also need to consider about Resource Description Framework (RDF) which is a directed graph and represents the web information. It details the data about the data or in other words the metadata and is something that is similar to the class diagrams.

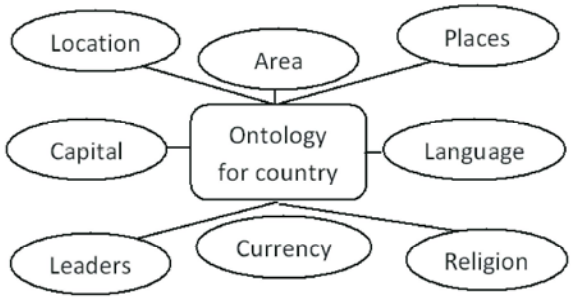

Fig 2: Ontology example – Country Objects

The present Universal Description, Discovery and Integration (UDDI) emphasis only on a single search criterion like the country name, currency, area, location or discovery URL. We can do this in a loop for searching multiple UDDI registries and then integrate all the results for publication using filtering methods. The registries discussed here unfortunately did not provide any additional service like checking the quality of the registered services because of which the results retrieved were not so good [13]. To find the quality of the registered services and to yield better retrieval results we use SVM with feedback mechanism which is detailed in the next section.

Web Services Classification Using SVM: Supervised learning refers to the task of understanding a function from labeled training data and Support Vector Machines are one such supervised learning model which along with the learning algorithms help analyzes data that can be used for classification and regression analysis.

Since the traditional classification methods typically require a medium to large sample collection, we have proposed to use SVM for web services discovery which uses the expressive information of groups in a large scale classification as sample data, so as to separate from the need on tester service documents. To do this, we need to first extract the features from the sample set and then train the system based on that instead of the original set itself which is generally huge.

The whole process starts with getting the document collection α = {d1, d2, d3… Dn}. Each document here corresponds to a domain or category C = {c1, c2, c3… cn} and a feature space F = {f1, f2, f3… fn}. The sample documents are identified and mapped to W = {w1, w2, w3… wn} where 'w' represents the weight of the document. The feature vectors are now fed as input to the SVM classifier to train the system. The documents are mapped as +/- 1 based on the relevancy during training. Once the system is trained, the unclassified or the test vector is fed to the classifier system and the output is predicted.The fig 3 below shows the complete process of web service discovery through classification and feedback mechanism.

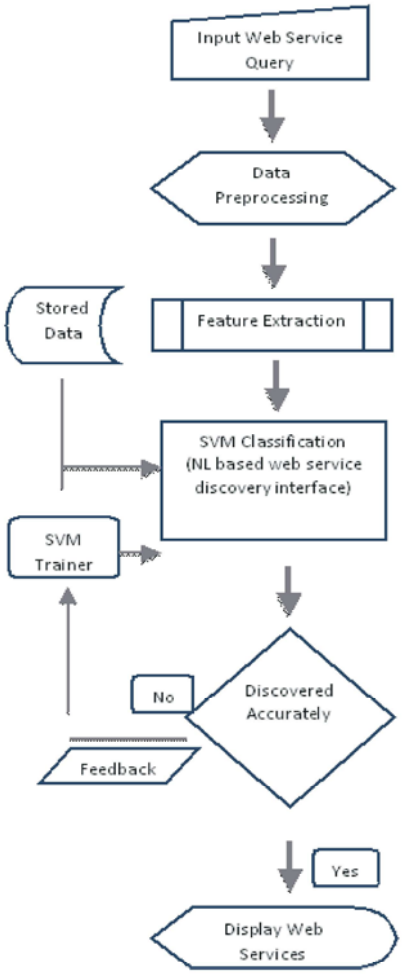

Fig. 3: Web services classification using SVM

When new web services gets registered, the input and output data will be extracted and mapped to a feature vector. This vector is further fed as input to the classifier in order to find the category or the domain, in our words it's the ontology. Hierarchical classification is usually preferred [11] to reduce the dimensionality.

Support Vector Machine algorithm outputs an optimal hyperplane which helps to classify new test vectors. The line that separates two classes is found and if there can be multiple lines drawn and then the algorithm tries to find an optimal one which helps classify the data in a better way. This separating hyperplane will try to maximize the distance of the training data as shown in fig 4 below.

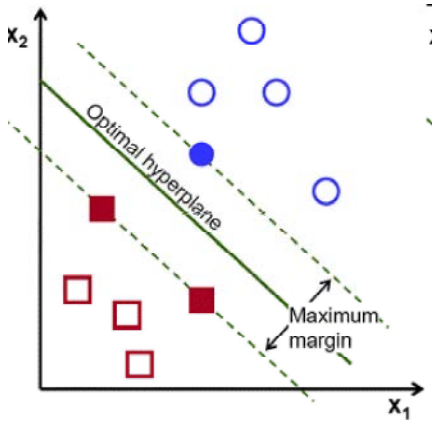

Fig. 4: SVM – Hyperplane representation

A hyperplane is a subspace of one aspect which is a smaller amount than its ambient space. The Classification that is discussed can be either linear or non-linear. Linear SVM [14] is represented as:

$$z_j = x_j^T w + b$$

where b represents a constant value and w is a vector which contains the weights of the d financial ratios. X represents the input set used for training and Z is the support vectors generated through the algorithm. The training data first gets normalized by finding the mean and standard deviation.

Z = (x – Mean) / standard deviation

Where Z represents the normalized value while x is the sample. Followed by this, a diagonal matrix (D) and error matrix (E) are created and finally an augmented matrix (A) is formed. This augmented matrix contains the normalized samples with the error matrix.

The aim of this SVM algorithm is to form a linear system that can adapt to the environment and to learn from experience for future classification. This is therefore an optimization problem where we need to find a vector of optimization variables (x1, x2, x3... xn) in order to minimize an objective function f(x) i.e.min f(x) subject to some constraints.

The resolution is given by the Lagrangian function as detailed below:

max (u) min (W, gamma) L (W, gamma, u)

Where W = Weight Vectors and the final decision function is given by

F(x) = sign value (W' x - gamma)

The equations below represent the actual problem variables including w, gamma, error and u in terms of the Lagrange multiplier as follows:

W = A'Du
Gamma= -e'Du
Error = (1/C) u where C = Learning rate (constant)

In the above set of equations derived, we know all the values except the variable u. The rest of the SVM algorithm below details on how to find this value. In mathematical optimization problem, two conditions namely (a) Karush–Kuhn–Tucker and (b) Sherman-Morrison-Woodbury are necessary conditions for anoptimal solution and they help in deriving the value of 'u' to be:

u = C(I -H ((I/C) + H'H)-1H')e

where

H = D[A -E]

The weight vectors also called as the support vectors are finally found along with the gamma value. When a new test vector comes for classification, we find its class by

f(x) = sign (W' x - gamma)

The accuracy of an SVM model in classifying a new test vector is largely reliant on on the range of the kernel constraints such as C, Gamma, P, etc. There are two methods available for identifying the optimal parameter

values which are a grid search and a pattern search. Grid methodefforts values of each constraint across the limited range by means of geometric steps. A pattern search on the other hand starts at the middle of the range specified and makes trial steps in every path for every constraint. When an improvement is observed, the questmidpoint changes to the new point identified and the procedure is reiterated. If there is no progress, the step size is condensed and the search is continueduntil it improves. When the search step size is reduced to a quantifiedacceptance, the pattern search stops.

In our web services classification problem, where we have multiple domains for training and testing, we can spread the discussed two-class linear classifiers to multiple classes or multiple domains as needed. The technique to use rest on whether the classes are mutually exclusive or not. Multilabel, or multivalue classification is followed for not mutually exclusive conditions. In this case, a document or a service can belong to multiple classes at the same time, or to a single class, or even to none of the classes formed. A verdict on one class will leave all the other choices open.

Improved Relevance Feedback Mechanism: Support Vector Machines are better than the traditional algorithms for web services retrieval when used in a relevancy feedback setting [15]. During web services retrieval, some of the results will be relevant while some are not. The relevancy here should be studied with user perception and not objectively.

Web services discovery starts with user presenting a query in natural language and the system returns the retrieved results as per the rank obtained. One screen of results is generally presented to the user and let us assumes that there were ten results for the user to decide on the relevancy. Depending on the quality of the input provided by the user, the retrieved results would have relevancy. If the services returned are relevant the user would proceed with that and if not he keeps scrolling to the subsequent pages until he finds the services matching his requirements.

When the user goes through the services, it gets marked as relevant results while those unmarked are classified as non-relevant at the first pass. The feedback continues until the user closes the procedure. Meanwhile, the support vector machine algorithm also tries to learn the reason for missing out the services that were interested by the user but still were not ranked at the top and tries to adjust the weight vectors of that particular domain accordingly.

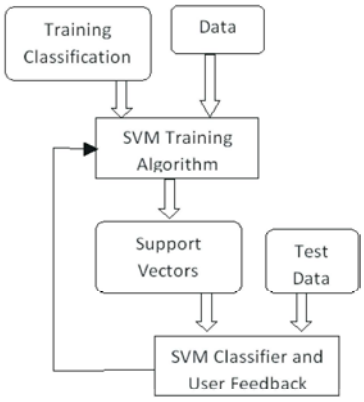

Fig. 5: Feedback Mechanism in improving the retrieval accuracy

The results can be classified in to one of the three categories including a) Nothing relevant, b) partially relevant and c) Irrelevant retrieved results. In the first case, we will have to wait until the user finds one relevant service from all the retrieved results. In the second case, only the relevant results are marked and ranked to the top. In the last case, we assume that the SVM algorithm is stabilized and the retrieval results are good. Fig 5 represents the feedback mechanism.

In relevancy feedback, we have to calculate the weight vectors and the constant values first and then followed by w*• D -b for all services that are not seen in the first set and rank them accordingly. If all the retrieved results are relevant to the input query, the weight vectors will not change while if the results are within the margin, then the next set of weight vectors will differ and get updated.

RESULTS AND DISCUSSIONS

Web services discovery begins with user providing the input query. Assuming that the database is already ready and sorted by their content and few other factors, we proceed with user query processing. This includes query understanding, spelling corrections, synonyms identification, search algorithm execution including classification and finally retrieve the top results for display.

There are a variety of mechanisms available for measuring the effectiveness of the proposed approach and one such commonly used metric is precision and recall. They are a measure of relevance and widely used in information retrieval process. Precision finds the results that are relevant to the query while recall measures the successful retrieval results from the set.

If R is used to represent the relevant services present, then nRel represents the number of relevant results retrieved using the proposed feedback methodology and N corresponds to the total set. Precision (P) and recall (r) are hence formulated as:

P = nRel / N and r = nRel / R

Table 1: SVM sample data from Nutrition source domain used for classification

Domain	Class A	Class B	Type
Carrot	-1.7984	-1.6730	1
Okra	-1.0791	-0.5937	1
Beans	-0.5995	0.7556	1
Eggplant	1.0791	-1.4032	1
Mushroom	0.1199	0.2159	1
Apple	0.3597	0.4857	-1
Blueberry	-0.3597	1.5651	-1
Fig	0.5995	0.4857	-1
Mango	0.1199	-0.3238	-1
Orange	1.5587	0.4857	-1

If the precision value decreases, it could either be due to algorithm efficiency or due to the absence of documents that are relevant to the input query. Hence we need to measure this data across iterations instead of a single iteration. Table 1 summarizes the precision results for a two class SVM trained to differentiate vegetable and fruits from Nutrition source domain.

Fig. 6: Precision-Recall curve

The test set considered here is from the nutrition source domain where the user query could be one of the following:

- Good fruits for reducing the cholesterol level present in the body
- Will apple help reduce the cholesterol level and help keep the body fit
- Is cabbage a healthy vegetable in terms of fat reduction
- Does blue berry contain antioxidant that can help reduce the lipid levels
- Which vegetables and fruits should I eat for reducing my bad cholesterol level in my body

Here the intention of different users is to find the best vegetables and fruits that can help control the cholesterol and keep the body fit. So, the domain classification along with the synonyms of different words like cholesterol, fat, lipid etc. are supposed to be grouped together.

We observe that once the training is done and when the user starts viewing the retrieved results, we start getting the feedback indirectly and update our support vectors accordingly. Once the machine learning is completed, we notice an improvement in precision-recall metrics and further updating weight vectors becomes inessential as shown in fig 6.

We have also tried to compare the advantages and disadvantages of support vector machine alongside maximum entrophy techniques. While SVM carry the advantages of being highly accurate and can handle many features, it lacks in speed and requires more time to process. Even training time along with the feedback mechanism takes a considerable amount of time for the algorithm to get stabilized and yield better classification results. On the other hand, Maximum Entropy techniques have a better speed and takes only a little time to process. Accuracy and efficiency of Maximum entrophy methods are low when compared to SVM. For these reasons, we stick on to support vector machines as they have the edge over the traditional algorithms available over the literature.

We have also tried it with few other sets from different domains including transportation, animals, food items etc. and each domain has provided us with the almost equal precision-recall metrics values. When we wanted to combine multiple domains and classify the user query to one of them, we find multi class SVM would be more suitable. It is built as ‘one-versus-all’ classifier and it chooses the class which satisfies the data set with the greatest margin among the rest.

CONCLUSION

With trillions of data and informationgetting published every year, the need for more user-friendly interfaces to retrieve the data effectively and efficiently are grows. In this paper, we have presented support vector machine based web services discovery along with the feedback mechanism which helps us with better retrieval accuracy. The feedback analysis system accepts the response from the users who provided the initial query and tries to adjust the weight vectors based on the retrieved results and user's interest.

We have also analyzed the performance of feedback mechanism based SVM algorithm with the maximum entropy methods. The regular methods results are not satisfactory and is not recorded hence. We also observe that when the database contains more relevant results the retrieval accuracy is higher among all algorithms and the results decrease when there is no sufficient data in the database or when the data in database is totally irrelevant to the user query. In those cases, our feedback mechanism based SVM gives us better web services discovery as compared to the rest of the algorithms. Hence we feel that the semantic gap that arises due to the difference between the intention of the user's query and the retrieved results gets minimized and bridged when we deploy feedback mechanism based support vector machine algorithm for web services discovery problem statement.

The high complexity of the algorithms along with the time taken to train and test the available data set is high with the proposed approach. Also the integration of solution in to the existing development environments is also difficult due to various factors. Our future directions would be towards solving these issues and come up with a more optimized solution for web services discovery.

REFERENCES

1. Wang Hongbing, Suxiang Zhou and Qi Yu, 2015. Discovering Web Services to Improve Requirements Decomposition, Proceeding of the 2015 IEEE International Conference on Web Services, 743-746.
2. Pablo Rodriguez-Mier, Carlos Pedrinaci, Manuel Lama and Manuel Mucientes, 2015. An Integrated Semantic Web Service Discovery and Composition Framework. IEEE Transactions on services computing, DOI 10.1109/TSC.2015.2402679.
3. Liang Jie-Zhang, 2012. Innovations, standards and practices of Web services: emerging research topics. Hershey, PA: Information Science Reference Publishers, Chapter 4,Web Services Discovery with Rough Sets.
4. Karagoz Funda, 2006. Application of schema matching methods Semantic Web service discovery. M.Sc. Thesis, The Graduate School of Natural and applied sciences of Middle East technical university.
5. Ethan Cerami, 2002. Web Services Essentials - Distributed Applications with XML-RPC, SOAP, UDDI & WSDL, O'Reilly Media publications.
6. Heather Kreger from IBM Software Group, 2001. Report on Web Services Conceptual Architecture, pp: 19-22.
7. Venkatachalam K., N.K. Karthikeyan and S. Kannimuthu, 2016. Comprehensive Survey on Semantic Web Service Discovery and Composition, International Conference on Engineering Technology and Science (ICETS'16)
8. Toch Eran, Iris Reinhartz-Berger, Avigdor Gal and Dov Dori, 2006. Bridging the Gap between Web Services and the Semantic Web, Next Generation Information Technologies and SystemsVolume 4032 of the series Lecture Notes in Computer Science, pp: 357-358.
9. Pavithra, G., D. Prabakar and Dr. S. Karthick, 2013. A Survey on query and keyword search in database, International Journal of Science, Engineering and Technology Research (IJSETR), 2(1).
10. Bouziane Abdelghani, Djelloul Bouchiha and Noureddine Doumiand Mimoun Malki, 2015. Question Answering Systems: Survey and Trends. The International Conference on Advanced Wireless, Information and CommunicationTechnologies (AWICT 2015)
11. Hongbing Wang, Yanqi Shi, Xuan Zhou and Qianzhao Zhou, 2010. Web Service Classification Using Support Vector Machine, 22nd IEEE International Conference on Tools with Artificial Intelligence.
12. Chen Zi Long and Yang Lu, 2011. Improving Relevance Feedback via Using Support Vector, Machines Advances in Civil Engineering, CEBM 2011.
13. Pautasso Cesare, 2004. JOpera: a toolkit for efficient visual composition of Web services, International Journal of Electronic Commerce.

14. Zhu Guangyu and Peng Zhang, 2014. Linear Programming v-Nonparallel Support Vector Machine. International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI).
15. Onoda, T., H. Murata and S. Yamada Zhu, 2006. Non-Relevance Feedback Document Retrieval based on One Class SVM and SVDD. The 2006 IEEE International Joint Conference on Neural Network Proceedings.