

Implementation of Rank Based Semantics Association for Service Discovery and Composition

T. Priyaradhikadevi¹, R. M. S. Parvathi², G.Ezhilarasi³

¹Research Scholar & Associate Professor
Department of Computer Science and Engineering
Mailam Engineering College, Mailam – 604304, Tamil Nadu, India
priyaradhikadevi@gmail.com

²Principal & Professor, Department of Computer Science and Engineering
Sengunthar College of Engineering, Tiruchengode – 637 205, Tamilnadu, India
drmsparvathi@gmail.com

³PG Scholar, Department of Computer Science and Engineering
Mailam Engineering College, Mailam-604304, Tamil Nadu, India
ezhilguna1989@gmail.com

Abstract: *Web services have become an incredible source for any internet technology development. Many web services are available within UDDI does not contain semantic descriptions so the appropriate user service request may not be obtained in service discovery process. In our paper we proposed a approaches for semantics service discovery for a given unclear description of service request. Service discovery involves semantics categorization and enhancement of service request. Semantics categorization is performed at offline in UDDI whereas enhancement performed at online which is the basis for achieving better matching of relevant services. We also proposed an algorithm for ranking the web services for a given service request and we showed experimental results for effective and feasible service discovery.*

Keywords: Semantic, service discovery, UDDI

1. Introduction

In Requirements Engineering (RE), Goal and Actor orientation has been recognized as an approach more promising than other system- and functionality-based techniques used in most of the traditional Software Engineering methodologies. By adopting the notions of Actor, Goal, and Intentional Dependency, it is in fact possible to refine high-level requirements originating from the organizational setting (i.e., stakeholders' needs and desires) into detailed descriptions of the system to be implemented (in terms of architecture, components, and functions), in a smooth and controlled manner.

A large number of web services structure a service oriented architecture and facilitate the creation of distributed applications over the web. These web services offer various functionalities in the areas of communications, data enhancement e-commerce, marketing, utilities among others. Some of the web services are published and invoked in-house by various organizations.

These web services may be used for business applications, or in government and military. However, this requires careful selection and Composition of appropriate web services. The web services within the service registry (UDDI) have predefined categories that are specified by the service providers. As a result, similar services may be listed under different categories. Given the large number of web services and the distribution of similar services in multiple categories in the existing UDDI infrastructure, it is difficult to find services that satisfy the desired functionality. Such service discovery may involve

searching a large number of categories to find appropriate services. Therefore, there is a need to categorize web services based on their functional semantics [1] [4] rather than based on the classifications of service providers. In order to address the limitations of existing approaches, an integrated approach needs to be developed for addressing the two major issues related to automated service discovery:

- a) Semantic-based categorization of web services
- b) Selection of services based on semantic service description rather than syntactic keyword matching.

2. Service Oriented Computing

The popularity of Web services preceded that of service-oriented computing. As a result, their initial use was primarily within traditional distributed solutions wherein they were most commonly used to facilitate point-to-point integration channels. As the maturity and adoption of Web services standards increased, so did the scope of their utilization.

With service-oriented computing comes a distinct architectural model that has been positioned by the vendor community as one that can fully leverage the open interoperability potential of Web services, especially when individual services are consistently shaped by service-orientation. For example, when exposing reusable logic as Web services, the reuse potential is significantly increased. Because service logic can now be accessed via a vendor-neutral communications framework, it becomes available to a wider range of service consumer programs.

3. Rank Semantic Association Algorithm

The complex relationships are based on property sequences that link the two entities in the semantic association [7] [8]. The rank Semantic association's algorithm as shown in provides the details of our approach. Two entities e_i and e_j are semantically associated with each other if there exists one or more relationship where $1 \leq i < n$ and $1 \leq j < n$. Next for each of these entities we find the relevance, specificity and the user-specified span. The user assigns weights for each of the parameters to refine the request. he also makes the ranking process more flexible. Our current approach assigns binary values to the ranking parameters. Assigning a range of specific values to these parameters is part of our future work. To illustrate our approach consider the following association pattern [2] {temperature, pressure, postal code}. The user-specified weights $w_1 = 0.3$, $w_2 = 0.4$ and $w_3 = 0.3$, coverage $\frac{1}{4} 2$, and depth $\frac{1}{4} 2$. The three concepts are linked to the concept Weather specified in the upper ontology. The concepts are located in the lower part of the concept hierarchy, this is indicative of greater specificity and as a result $Sp = 1$. Next we determine if the concepts fall within the specified span within the weather domain. As illustrated by the Weather Concepts ontology, the concepts are included in the specified span thus $S = 1$. The semantic rank score of the association pattern is calculated as $0.3 \times 0.4 \times 1 \times 0.3 \times 1 \times 1$. The associated semantic rank [6] is utilized to sort the association a pattern collection.

Algorithm: rankSemanticAssociations
Input: Association Pattern Collection (P) formed in Hyperclique pattern mining phase
 User assigned weights $\{k_1, k_2, k_3, coverage, depth\}$,
Output: Semantic rank score ScoreSemRank
 1: begin
 3: for each association pattern $p(x,y) \in P$ do
 4: Traverse H_c for upper ontology concept C
 5: if $p(x,y) \subseteq C$
 6: assign $Rel_i = 1$
 7: Calculate $[depth\{p(x,y) \mid C\}]$
 8: if $depth\{p(x,y)\} > 5$
 9: assign $Sp_i = 1$
 10: For each term in $p(x,y)$
 11: calculate $coverage_i(p(x,y))$
 12: calculate $depth_i(p(x,y))$
 13: if $coverage_i \leq coverage$ && $depth_i \leq depth$
 14: assign $S_i = 1$
 15: $ScoreSemRank = k_1 * Rel_i + k_2 * S_i + k_3 * Sp_i$
 16: end for
 17: return ScoreSemRank
 18: end

4. Requirement Engineering On Service Oriented Computing

Requirement engineering framework for SOA development consists of four activities such as Service Identification, Service Specification, Service Realization and Service Deployment as shown in Fig-2. Service Identification (SI), point to identification and selection of candidate services; Service Specification (SS) refers to Specification of the set of services by developing a Service Model; Service Realization (SR), refers to the designing of service component in design model and finally Service Deployment (SD refers to the transferring of the services to the production environment.

5. Semantic Categorization of Web Services

In our approach we begin with the semantic categorization of UDDI wherein we combine ontologies with an established hierarchical clustering methodology, following the service description vector building process. For each term in the service, description vector, a corresponding concept dislocated in the relevant ontology. If there is a match, the concept is added to the description vector. Additional concepts are added and irrelevant terms are deleted based on semantic relationships between the concepts. The resulting set of service descriptions is clustered based on the relationship between the ontology concepts and service description terms. Finally, the relevant semantic information [2] is added to the UDDI for effective service categorization. With respect to our running example, additional concepts from weather ontology are added to the description vectors for WS1 and WS2. Following this, both WS1 and WS2 are grouped together utilizing hierarchical clustering. All the services within this cluster (including WS1 and WS2) are then associated with an upper ontology concept "weather" as a category. Below is an outline of the key steps of our approach as

- Build the web service description vectors.
- Append relevant ontology concepts and delete irrelevant terms based on the ranking of semantic relationships among the terms.
- Mine web service collection utilizing hierarchical clustering and associate an upper ontology concept for each cluster and the relevant ontology concept for the corresponding sub cluster.

5.1 Parameters-Based Service Refinement

The next step is service selection from the relevant category of services using parameter-based service refinement. Web service parameters, i.e., input, output, and description, aid service refinement through narrowing the set of appropriate services matching the service request. The relationship between web service input and output parameters may be represented as statistical associations. These associations relay information about the operation parameters that are frequently associated with each other. To group web service input and output parameters into meaningful associations, we apply a hyper clique pattern discovery approach. These associations combined with the semantic relevance are then leveraged to discover and rank web services. For the running example, the first step of our approach for parameters-based service refinement is to build the service parameters association pattern item set for all services within the "weather" cluster (including WS1 and WS2). The next step involves pruning the association pattern based on concepts extracted from domain ontology and a confidence threshold. This provides a set of ranked web services matching service functionality.

- Retrieve associated parameters forming the association pattern item set.
- Perform hyper clique pattern discoveries on the association pattern item set.
- Rank the semantic associations between the terms

- d. Prune the association patterns collection.

5.2 Semantic Similarity-Based Matching

The parameter-based refined set of web services is then matched against an enhanced service request as part of Semantic Similarity-based Matching. A key part of this process involves enhancing the service request approach for web semantic similarity-based service selection employs ontology-based request enhancement and LSI based service matching. The basic idea of the proposed approach is to enhance the service request with relevant ontology terms and then find the similarity measure of the semantically enhanced service request with the web service description vectors generated in the service refinement phase. For evaluating this similarity, we employ LSI-based technique that uses cosine measure as the similarity metric. A key issue in discovery of web services refers to the query language utilized to form the web service request. The web service request can be formed in two ways, i.e., a syntactic web service request and a semantic web service request. The syntactic web service request, in its most basic form, utilizes simple text to form a web service request. Syntactic web query languages such as XQuery, XSLT, GQL, and Lucene among others, which have been tailored specifically for declarative and efficient access and processing of web data, may also, be utilized to form the web service request. The web service request may also be formed of a set of semantics-based XML languages, such as RDF and OWL, that rely on ontologies to explicitly specify the content of the tags to annotate the service request. Most of the RDF query languages today are relational based, such as SPARQL, RQL, and TRIPLE among others. Compared with formal queries, keyword-based queries have the following advantages:

- a) A simple syntax in terms of a list of keyword phrases,
- b) Open vocabularies wherein the users can use their own words to express their Information requirement, and
- c) The familiarity of the user with these interfaces due to their widespread usage. However, the fundamental disadvantages of a keyword based web service request are the lack of precision and the lack of verifiability.

A new, semantics-based approach is necessary not only to reduce this information overload problem, but also to enable more effective and productive services over the web. Our research validates the limitations of keyword-based searching and provides an approach in which semantics enhanced web service request overcome these limitations. In this paper, we report on the experiment in which we evaluate the benefits and drawbacks of the added value and pitfalls of semantic enhancement of web service request over pure keyword matching technique. Thus, though keyword-based web service discovery has proven its usefulness, applying semantics-based web service request strategies should greatly increase the resulting precision of searches and enable new types of web service requests to be formed. For our running example, we use the weather web service request: Service Request (SR). Find the temperature and rainfall based on

zip code. Below is an outline of the key steps of our approach), followed by a detailed discussion of each of the steps.

- a. Preprocess service request and determine the overall search category of web services for the search.
- b. Index the web service description collection and retrieve relevant service descriptions.
- c. Preprocess the service descriptions set and retrieve associated concepts related to the initial service request from the ontology framework.
- d. Acquire the associated concepts related to the initial service request to expand the request. Transform the service description set into a term-document matrix.
- e. Perform SVD on this matrix.
- b) Project the description vectors and the request vector and utilize the cosine measure to determine a similarity.

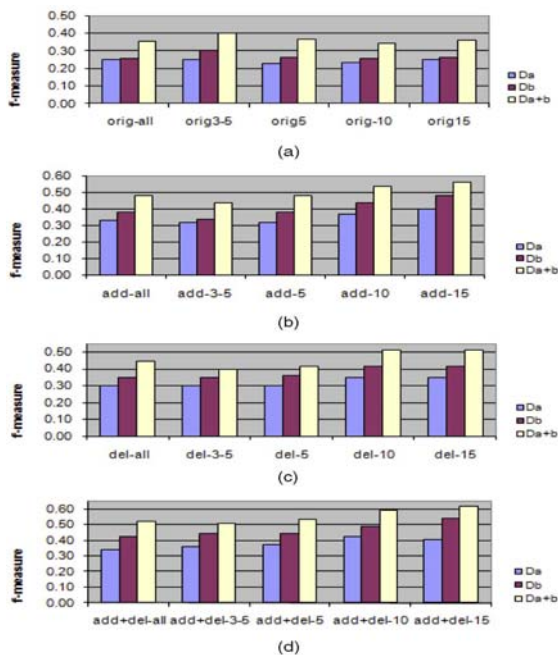
6. Experimental Results

Experiment 1: Original Setup

Depicts the results for the original setup, As observed in all the experiments the f-measure values are far from 1. The experimental results serve as a baseline for comparing the results with other data setups.

Experiment 2: Add Setup

Shows the results for the Add setup, we observe that adding relevant terms from ontology yields an improvement over experiments conducted with the original data sets as illustrated in Fig. 10b. This leads us to a conclusion that adding relevant domain knowledge for all the terms is not all that helpful. The lack of high returns in results is on account of the generic nature of the SUMO ontology that does not focus on a specific domain. This may be due to the fact that a large number of web service descriptions have overlapping categories. The addition of terms related to these overlapping domains creates additional noise which is not resolved by the clustering algorithm. A possible approach to overcome this effect would be to consider addition of concepts from the ontology to only the relevant terms, accounting for context. The ontology serves as a guide for clustering that incorporates domain knowledge and more focused information. We consider two criteria viz., span and depth, to determine the coverage of the ontology concepts. The exact parameters determining the coverage aim to achieve the smallest set of additional ontology concepts while maintaining the best overall coverage within the smallest set.



Experiment 3: Delete Setup

Better results for cluster quality were observed with term reduction from the service description vectors as illustrated in Fig. 10c. The term reduction involved pruning individual term vectors of irrelevant and low frequency terms which increases the specificity of the services.

Experiment 4: Add and Delete Setup

This setup aims to maintain a balance between the generality and the specificity of terms in web service Descriptions. This is achieved by expansion of the term vectors with relevant ontology concepts and subsequent reduction of terms from the web service descriptions. The results follow those observed in the add set of experiments. The technique, where in ontology concepts are added to all terms of web service descriptions followed by pruning, results in increased generality. The best results compared to all techniques were observed in the technique, where in ontology concepts are added to relevant terms of web service descriptions followed by pruning. This results in an increase in specificity and reduction of generality of the terms in web service description. The improved results may be explained on account of the generality specificity balance achieved by added semantic providing a good representative set for better categorization and the overall reduction of noise added to the vector representations.

7. Summary Results

Our evaluation deals with the frequency of service categorization for the entire UDDI. We perform service categorization on an incremental basis. We assume that the ontology is not perfect and that the ontology is updated to represent additional domain objects and their interrelationships. Then the categorization must be performed every time a newer service is added to the UDDI. However, periodic categorizations may be required if the service additions are frequent, as can be expected in real-life situations with large user and provider

communities. However, we can update the service category by isolating the upper ontology concept that remains unchanged and then recategorizing all the services that fall in its child concepts. When evaluating the efficiency of our approach, there are a number of factors that affect the timings obtained viz., the size of the underlying ontology and the number of service to be categorized. We found that the total processing time for the service categorization was 259 seconds for our test set of 800 web services with an approximate 1,000 concepts of the ontology data.

Web Service(WS)	Enhanced Service Request	Service Request
Identification	Score	Score
WS1	4.8496	4.3231
WS2	1.3798	1.0426
WS3	9.0672	2.9231
WS4	1.8802	0.9957
WS5	3.2128	1.0405
WS6	9.9376	3.5060

Web service scores for individual weather services

References

- [1] M.A. Corella and P. Castells, "Semi-Automatic Semantic-Based Web Service Classification," Proc. Int'l Conf. Business Process Management Workshops (BPM '06), 2006.
- [2] P.W. Foltz and S.T. Dumais, "Personalized Information Delivery: An Analysis of Information Filtering Methods," Comm. ACM, vol. 35, no. 12, pp. 51-60, 1992.
- [3] E. Han, G. Karypis, and V. Kumar, "Scalable Parallel Data Mining for Association Rules," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '97), 1997.
- [4] A. Heb and N. Kushmerick, "Automatically Attaching Semantic Metadata to Web Services," Proc. IJCAI Workshop Information Integration on the Web, 2003.
- [5] http://dit.unitn.it/~kokash/documents/WS_matching-hybrid.pdf, 2012.
- [6] Lei, Y., An Instance Mapping Ontology for the Semantic Web. In Proceedings Third International Conference on Knowledge Capture (KCAP2005).
- [7] McIlraith, S., Son, T., and Zeng, H. Semantic web services. IEEE Intelligent Systems 2001.
- [8] Deerwester, S., Dumais, S. T., Landauer, T. K., Furnas, G. W. and Harshman, R. A., "Indexing by latent semantic analysis." Journal of the Society for Information Science, 41(6), 391-407.