Semantics-based Web Service Discovery Using Information Retrieval Techniques

J.Hou, J.Zhang, R.Nayak, A.Bose

Faculty of Information Technology, Queensland University of Technology

jun.hou@student.qut.edu.au,
{jinglan.zhang, r.nayak, a.bose}@qut.edu.au

Abstract. This paper demonstrates an experimental study that examines the accuracy of various information retrieval techniques for Web service discovery. The main goal of this research is to evaluate algorithms for semantic web service discovery. The evaluation is comprehensively benchmarked using more than 1,700 real-world WSDL documents from INEX 2010 Web Service Discovery Track dataset. For automatic search, we successfully use Latent Semantic Analysis and BM25 to perform Web service discovery. Moreover, we provide linking analysis which automatically links possible atomic Web services to meet the complex requirements of users. Our fusion engine recommends a final result to users. Our experiments show that linking analysis can improve the overall performance of Web service discovery. We also find that keyword-based search can quickly return results but it has limitation of understanding users' goals.

Keywords: Web service discovery, Semantics, Latent Semantic Analysis, Linking Analysis

1 Introduction

With the popularity of Service Oriented Architecture (SOA), many enterprises offer their distributed Web services as interfaces for their core business systems. Web services are embracing unprecedented attention from the computer world. Web services can be discovered by matchmaking requirements of service requesters with providers. Web service discovery plays a key role in finding appropriate Web services. Although a number of Web services are provided by different organizations, there are still no standards for Web service design and provision. Many Web services have the same or similar functionalities but they are described in various ways. It is a challenging task to discover accurate Web services in accordance with users' requirements.

Web Service Description Language (WSDL), the standard description language, and Universal Description Discovery and Integration (UDDI) for advertising Web services, are introduced to discover and invoke Web services. Web services requesters and providers then communicate with each other by SOAP message, a XML format communication language based on HTTP. The WSDL and UDDI search mechanism

utilizes syntactic search based on keywords because it can return a large number of Web services in a relatively short time. However, exact keyword match may miss actually relevant services and semantic search is proposed to enhance the search accuracy. In addition, since different organizations design services in enclosed circumstances, atomic Web services cannot satisfy different users' requirements [13]. One service can invoke other services to achieve goals with complicated requirements. Therefore, a set of Web services need to be composed to fulfill given tasks.

Two methods are used in this paper, namely Latent Semantic Analysis (LSA) supported by Wikipedia corpus and BM25 supported by WordNet. Wikipedia corpus is used to create Latent Semantic Kernel, while WordNet is introduced to improve the search performance of BM25. On top of that, we propose a linking analysis, which can automatically compose possible atomic Web services to conduct user-preferred tasks. In the fusion engine, a new result with atomic and composite Web services is recommended to users.

2 Related Work

This section summarizes some previous work in Semantic Web service discovery.

Due to the lack of semantics in WSDL, many semantic Web service description languages such as OWL-S, WSMO and WSDL-S have emerged to explicitly annotate WSDL with semantic information. OWL-S and WSMO demonstrate Web services semantics at a distinct level [12]. OWL-S is more concentrated on the "Upper ontology" (not domain-specific ontology) for describing Web services [3]. Compared to OWL-S, WSMO is more focused on producing a reference implementation of an execution environment, the Web Service modeling execution environment and specifying mediators [13]. Mediators are not the significant consideration in OWL-S conceptual and implementation [17]. However, the discovery mechanism in WSMX is based on keyword and simple semantic description [17]. Compared to OWL-S, WSDL-S has several advantages over OWL-S. First, details of both the semantics and operations can be described in WSDL. In addition, the semantic domain models are detailed externally, which offers Web service developers an opportunity to select the preferred ontology language. On top of that, the existing tool can be updated relatively easy. The objectives of WSDL-S are to be of compatibility with OWL-S with emphasizes on a more lightweight and incremental approach [14]. Although more lightweight and flexible (supporting different ontologies) ontology languages are emerging, there is still no standard ontology and the maintenance cost is very high with low scalability.

Many researchers make use of traditional Information Retrieval techniques. They parse WSDL documents into bags of words and create a term-document matrix. Then Webs services are ranked by Term Frequency–Inverse Document Frequency (TF-IDF) according to the term frequency of search query in each document. A binning & merging-based Latent Semantic Kernel [2] is proposed to enhance the semantics of LSA. The experiment result shows that the LSA approach can be acceptable both in scalability and complexity [19]. A method using surface parsing of sentences to add

structural relations [4] are proposed to improve the performance on single sentences in LSA. However, there still are some issues related to LSA. The pre-process including stop word removal and stemming reduces common terms and outliers, it also breaks WSDL structure at the same time. Nayak & Iryadi [15] and Hao & Zhang [7] propose Schema matching approaches in WSDL-based Web service discovery. Such approaches try to find not only text but also structure information for comparing WSDL documents. To effectively investigate semantics in text, a Wikipedia-based structural relationship-enhanced concept thesaurus [8] is introduced. This approach concentrates on improving the semantic relationships between important terms by applying text clustering. Above approaches are only keyword-based and simple keywords may not represent the preference of users very well. Users' selection of services is highly impacted by non-functionality such as response time, price, throughout, availability, reliability etc.

Researchers are devoted to dig more semantic information from current Web resources. Ding, Lei, Jia, Bin, & Lun [5] propose a discovery method based on Tag. Tags are widely used in images, bookmarks, blogs and videos to annotate the content of them. This approach suffers the same problem of above ontology languages. It is limited by the scope of comment on Web services and the variety between different comment styles. Semantic Web Services Clustering (SWSC) [16] makes use of preconditions and effects from OWL-S to improve the accuracy of Web service discovery. Using translation tools, more context information such as preconditions and effects after invocation can be collected thereby increasing the consistency of Web service discovery. In this method, hidden Web services can be discovered and be attached to similar groups before conducting search. However, scalability is still a problem.

3 Discovery approach

We propose a novel three-phase approach for Web service discovery. Figure 1 shows an overview of this methodology. In the semantic analysis phase, there are two methods used to retrieve atomic Web services, namely Latent Semantic Analysis (LSA) supported by Wikipedia corpus and BM25 supported by WordNet. Before applying those approaches, standard text pre-processing is performed to parse WSDL documents into bags of words. During this stage, stop word removal and stemming have been executed.

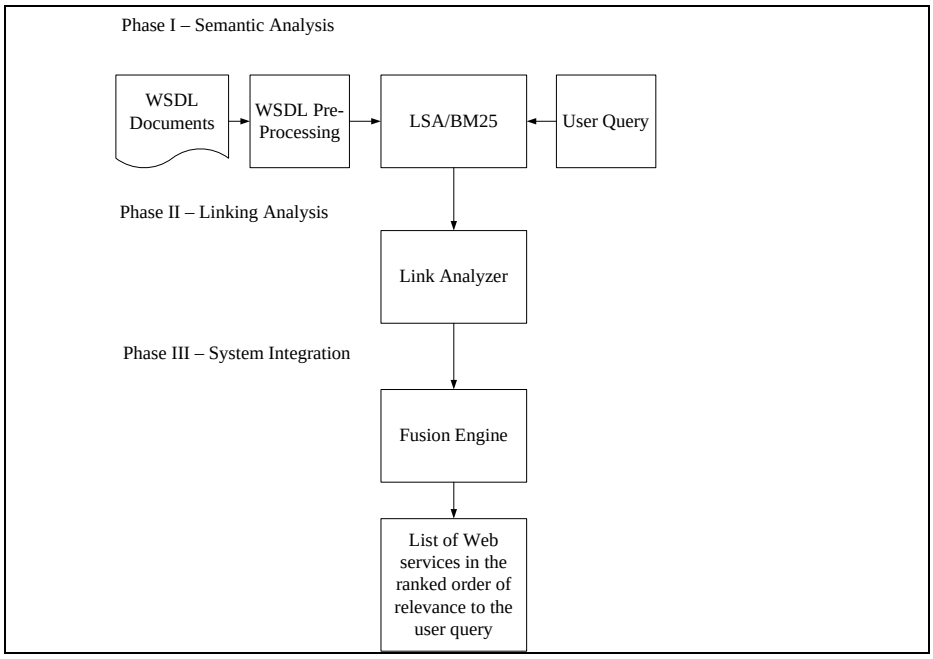

Fig. 1. Overview of Web Service Discovery Methodology

3.1 Pre-Processing

Stop word removal aims to reduce words which act poorly as index terms. For example, those words can be “a”, “the”, “and” etc. An external stop word list is introduced to filter out those words to perform data analysis.

Stemming is a process to replace words with their root or stem forms by removing affixes (suffixes or prefixes). Words such as “computing”, “computer” and “computed” will be replaced by the word “compute”. This process reduces not only the variety of words also the computation cost. The Porter Stemming Algorithm [18] is used to conduct the stemming process.

3.2 Semantic Analysis

Latent Semantic Analysis (LSA)

Figure 2 shows the overview of Latent Semantic Analysis (LSA). In LSA, the semantic kernel is used to find semantic similarity between Web services and users’ queries. The semantic kernel is constructed from a general-purpose dataset. The wikipedia dataset [2] is chosen because it is not domain-specific and covers various topics. Figure 2 shows the overview of LSA in phase I.

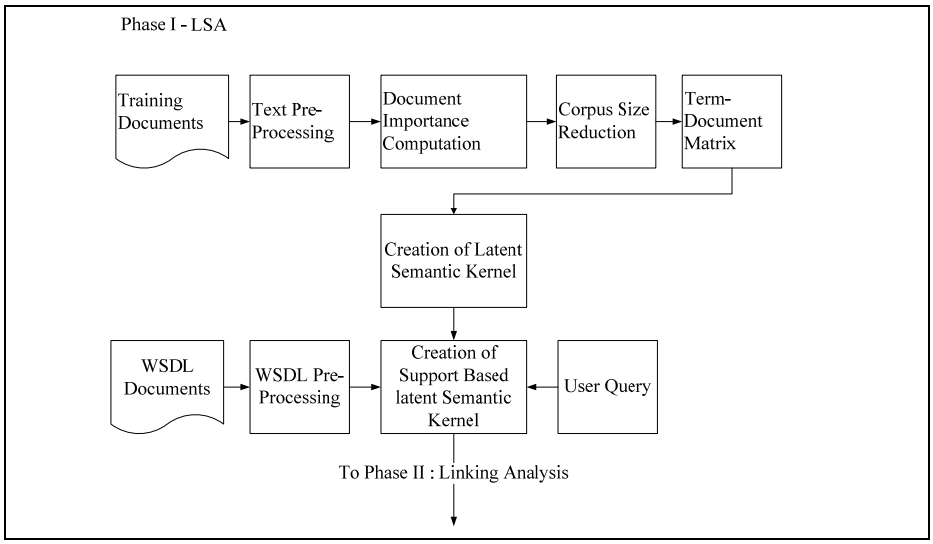

Fig. 2. Overview of LSA

To start, each pre-processed WSDL document is then encoded as a vector. Components of the vector are terms in the WSDL document. Each vector component reflects the importance by TF-IDF. The user query is also converted to a vector which is compared with the vector of a WSDL document. The similarity between the user query (Q) and the Web service document (W) is represented by the cosine value of two vectors. Equation 1 shows how to calculate the similarity between Q and W.

$$\text{Sim}(Q, W) = \text{Cos}(Q, W) = \frac{Q \cdot W}{\|Q\| \|W\|} \quad (1)$$

However, we use semantic kernel (K) here to enhance the semantics between Q and W. The Q and W is replaced with $Q^T K$ and $K^T W$ respectively. Equation 2 shows the improved equation with semantic kernel.

$$\text{Sim}(Q, W) = \text{Cos}(Q, W) = \frac{Q^T K \cdot K^T W}{\| Q^T K \| \| K^T W \|} \quad (2)$$

Finally, the top-k Web services are returned to users (k is set to 20).

BM25

BM25 is a bag-of-words retrieval algorithm that ranks documents based on the query terms appearing in each document. To increase the amount of query terms, WordNet is introduced to incorporate with BM25. WordNet is a general ontology, which can boost semantics from users' queries. Figure 3 shows the overview of using BM25 in phase I.

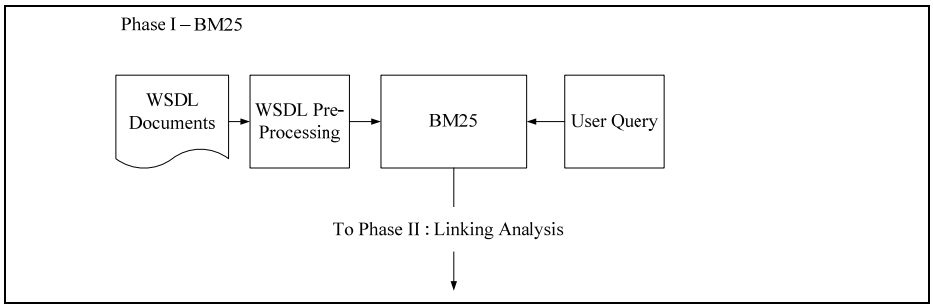

Fig. 3. Overview of BM25

After pre-processing, WSDL documents (W) are computed with users' queries (Q) for similarity. Equation 3 shows the major equation of BM25.

$$\text{Score}\,(Q,W) = \sum_{i}^{n} \text{IDF}\,(q_i) \cdot \frac{f_{(q_{i,W})} \cdot (k_i + 1)}{f_{(q_{i,W})} + k_i \cdot (1 - b + b \cdot \frac{|W|}{\text{avgdl}})} \qquad (3)$$

In $f_{(q_{i,W})}$, q_i is the term frequency in the WSDL document W. |W| is the length of the WSDL document and avgdl is the average document length in the text collection. Equation 4 shows the details of IDF (q_i).

$$\text{IDF}\,(q_i) = \log \frac{N - n_{(q_i)} + 0.5}{n_{(q_i)} + 0.5} \qquad (4)$$

N represents the total number of WSDL documents in the collection and $n_{(q_i)}$ is the number of WSDL documents containing q_i.

Same as LSA, the top-k Web services are returned to users (k is set to 20).

3.3 Linking Analysis

Web services are retrieved based on the query of a user. However, one Web service may not meet the requirement of the query of a user. For example, the query from a user is "weather by postcode" and the actual Web service is "weather by location". Obviously, the Web service needs to corporate with another Web service such as "postcode to location". Linking analysis aims to link possible Web services to satisfy the requirements of users. Figure 4 shows the overview of linking analysis.

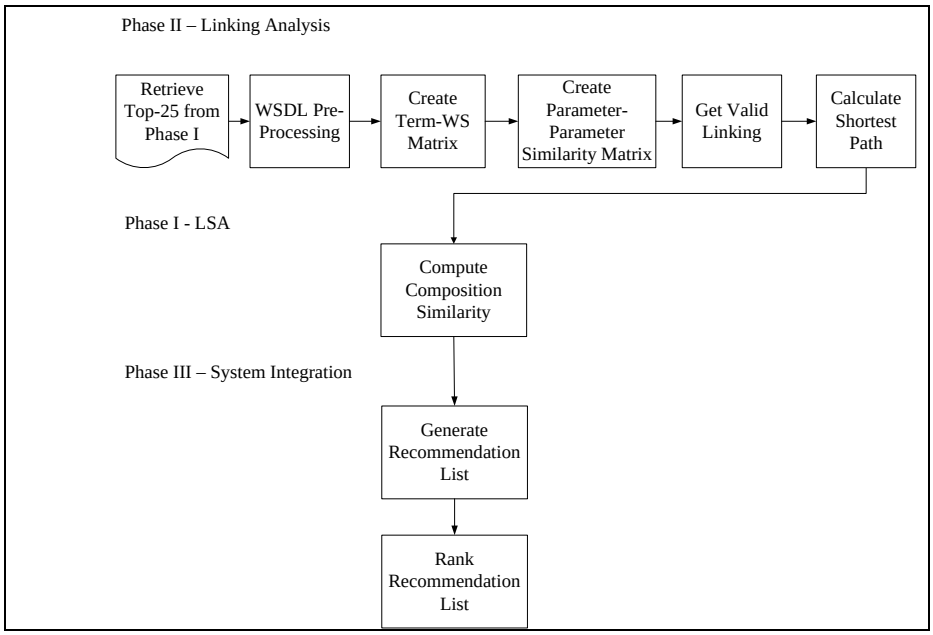

Fig. 4. Overview of Linking Analysis

In linking analysis, we use top 25 results from LSA or BM25 instead of directly linking Web services in the collection. In a WSDL document, the <PortType> tag consists of sets of <Operation> tags, which contain the description of invocable functions. We consider that Web services can be linked together if one Web service's output parameters match another one's input parameters in parameter name, parameter amount and data type. That information of input and output parameters is extracted for linking analysis. During the extraction, non-topic words such as "result" and "response" are filtered out.

Once we get parameter names, they are decomposed into tokens. For instance, "ChangePowerUnit" is split into "Change", "Power" and "Unit" from each capital letter. If two parameter tokens are exactly same, we consider it as exact match. However, there are parameters having tokens such as "car" and "vehicle" and they semantically can be linked. Therefore, we calculate the similarity between input and output parameters to semantically link two Web services. Equation 5 shows how to compute the similarity of two parameters.

$$\text{Sim}\ (P_1,\ P_2) = \frac{n \cdot \text{Sum}(\text{sim}(W_{P_1}, W_{P_2}))}{N} \qquad (5)$$

$\text{Sum}\left(\text{sim}\left(W_{P_1}, W_{P_2}\right)\right)$ is the sum of the similarity between tokens in parameter$P_1$and P_2 . N represents the total number of parameter tokens in P_1 and P_2. n is the minimum of the number of parameter tokens in P_1 and P_2. For example, if P_1 has 2 tokens and P_2 has 3 tokens, n will be 2 and N will be 5. If $\text{Sim}\ (P_1,\ P_2)$ is greater

than 0.98, we consider that the two parameters can be linked (linkable parameters). Furthermore, we use another factor, link strength, to decide if the two Web services can be linked. Link strength demonstrates the compatibility of two Web services by the number of linkable parameters. Equation 6 shows how to calculate the link strength.

$$\text{Link Strength} = \frac{N_L}{N_I} \quad (6)$$

N_L is the total number of linkable parameters and N_I is the number of input parameters of one Web service. Once we have the link strength, functions of Web services are converted to a graph where nodes representing functions are connected with each other by link strength. Afterwards, we use Floyd Warshall algorithm [6] to calculate the shortest path from each method to all other methods. We define composition strength as the average of link strength of a composition. All compositions are ordered by composition strength. Each composition is treated as a new Web service and compared with users' queries for similarity by LSA.

3.4 System Integration

The main purpose of integration is to integrate the results from composition of Web services with the atomic ones from LSA or BM25. The most important task is to decide which result appears in the final list. Generally, composition result has a higher accuracy than an atomic one. In addition, if a Web service is the component of a composition, it will not appear in the final result. Therefore, we select all compositions to the final result and then add atomic results to form top 20 recommendations. Figure 5 shows the overview of System Integration.

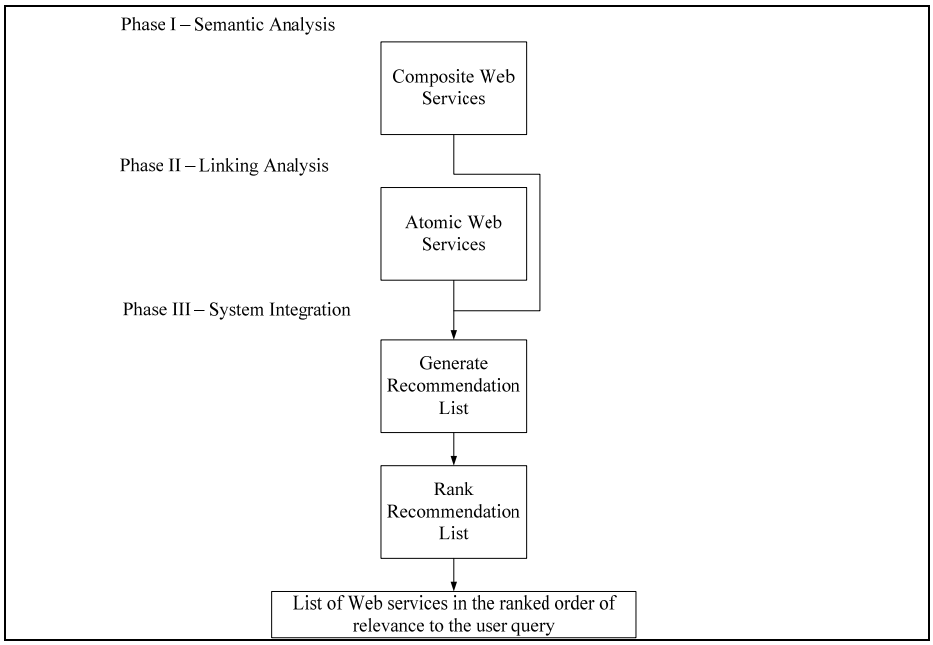

Fig. 5. Overview of System Integration

4 Data Set

The document collection is provided by the INEX 2010 organizing committee. The dataset [11] contains over 1,700 documents in the format of WSDL 1.1, which are directly crawled from real-world public Web services indexed by the Google search engine.

5 Evaluation

There are 25 topics from different domains. User queries are created by competition participants to ensure the variety. Figure 6 demonstrates the precision & recall curve under the query term “map”.

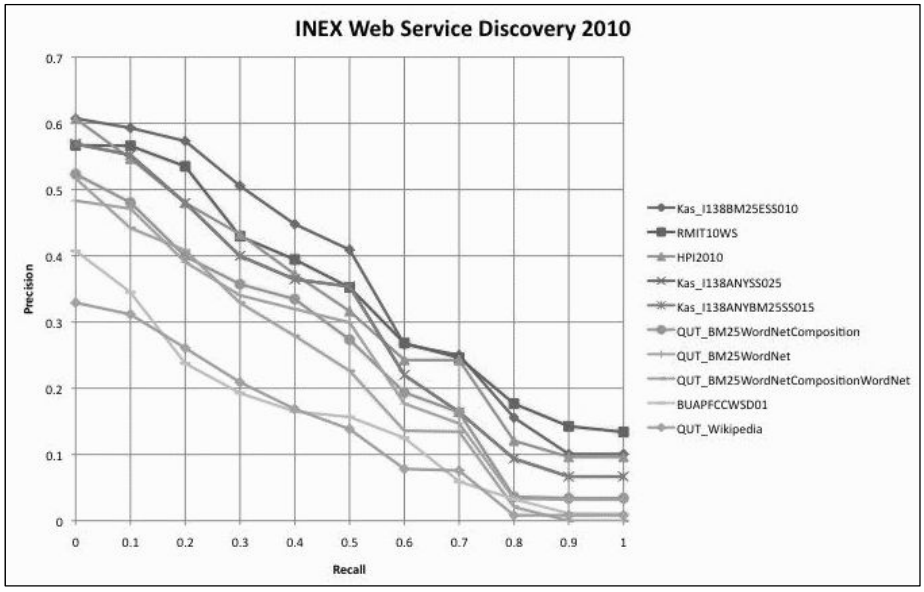

Fig. 6. INEX Web Service Discovery Results [10]

In Figure 6, we have four runs with the initial QUT and the best of them is QUT_BM25WordNetCompositon. In the BM25WordNetComposition run, we use BM25 supported by WordNet and the linking analysis. This submission outperforms BM25WordNet, which only applies BM25 supported by WordNet. It suggests that the linking analysis improves the accuracy of Web service discovery. The submission QUT_Wikipedia is created using LSA and it does not perform very well. One reason may be that the query term, "map", is simple and LSA cannot find semantic services. Another reason might be that the semantic services found by LSA are not closely relevant to the query term.

As we can see from Figure 6, the run Kas_I138BM25ESS010 (from Kasetsart University) has the highest precision when recall is less than 0.6. According to the result released from INEX, Kas_I138BM25ESS010 has the highest score, which is 0.3469 [10] with the query term "map". Kasetsart University makes use of pre-processing techniques to boost search accuracy [9]. In our method, we only use simple pre-processing techniques due to time-constraint. Our approach can be improved by applying sophisticated pre-processing techniques. Our linking analysis is proposed under the situation of multiple query items. For example, if a user types "weather by postcode", a combination of services "weather by city" and "city to postcode" will be retrieved. As a result, it almost has no effect with the simple query "map". We believe our algorithm will have better performance for more complicated multiple term queries, especially for queries that can only be satisfied with the linking of multiple atomic services.

Figure 6 has also shown that even by utilizing sophisticated pre-processing techniques, the score is only 0.3469, which means there are still a lot of irrelevant retrievals. That is because textual information occupies only a small part of the overall

size of WSDL documents so it is not sufficient enough to answer service queries on the single basis of query terms [1]. Moreover, WSDL documents describe the interfaces of Web services in an abstract way. The small size of textual information makes it relatively difficult to understand what the service offers. For example, sometimes, the service having a service name “map” in the service name tag may provide the map of a hospital or the map of a city. Keyword-based search with simple query term may not be suitable enough to answer users’ queries.

6 Conclusions and Future Work

The experiment result shows that both the Latent Semantic Analysis and BM25 boosted by WordNet approaches work for web service discovery. BM25 outperforms the Latent Semantic Analysis approach. Link analysis automatically composes Web services to fulfill complex tasks. Unfortunately this cannot be demonstrated by simple queries that can be satisfied by atomic services.

In this paper, Web services are converted to bags of words and then compared with users’ queries for similarity. However, we find that WSDL documents are not like normal documents having high richness of terms. More decomposition rules are needed to deal with abbreviation and artificial names when parsing WSDL documents. Furthermore, WSDL describes services in an abstract way sometimes just a single term in one tag. The single term cannot describe the function very well and investigating semantics by single words may cause more false negatives by misunderstanding the service functionality. In addition, discovering web services by considering only query terms is overly simple because Web services are involved in more complex business scenarios. Service choreography and service orchestration are considered when deploying and invoking Web services. Web services contain more business relationships than normal documents, especially during invocation. Non-functional parameters such as response time, price, throughout, availability, reliability etc have become significant factors on selecting services. As a result, more practical situations need to be investigated to effectively retrieve and select Web services.

References

1. Al-Masri, E., & Mahmoud, Q. H.: Identifying Client Goals for Web Service Discovery. In: 2009 IEEE International Conference on Services Computing, pp. 202--209. Bangalore, India (2009)
2. Bose, A., Nayak, R., & Bruza, P.: Improving Web Service Discovery by Using Semantic Models. In: 9th International Conference on Web Information Systems Engineering, pp. 366--380. Auckland, New Zealand (2008)
3. Burstein, M. H., & McDermott, D. V.: Ontology Translation for Interoperability among Semantic Web Services. AI Magazine. 26, 71--82 (2005)
4. Dennis, S.: Handbook of Latent Semantic Analysis. Mahwah, N.J (2007)
5. Ding, Z., Lei, D., Jia, Y., Bin, Z., & Lun, A.: A Web Service Discovery Method Based on Tag. In: 2010 International Conference on Complex, Intelligent and Software Intensive Systems, pp. 404--408. Krakow, Poland (2010)

6. Floyd, Robert W.: Algorithm 97: Shortest Path. Communications of the ACM. 5, 345(1962)
7. Hao, Y., & Zhang, Y.: Web Services Discovery Based on Schema Matching. In: 13th Australasian Conference on Computer Science, pp. 107--113. Ballarat, Australia (2007)
8. Hu, J., Fang, L., Cao, Y., Zeng, H.-J., Li, H., Yang, Q., et al.: Enhancing Text Clustering by Leveraging Wikipedia Semantics. In: 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 179--186. Singapore, Singapore (2008)
9. INEX 2010 Workshop Pre-proceedings, http://inex.otago.ac.nz/data/publications.asp
10. INEX Web Service Discovery Results, http://goanna.cs.rmit.edu.au/~jat/INEX2010_WS_results.html
11. INEX 2010 Web Service Track, http://www.inex.otago.ac.nz/tracks/webservices/webservices.asp
12. Lara, R., Roman, D., Polleres, A., & Fensel, D.: A Conceptual Comparison of WSMO and OWL-S. In: L.-J. Zhang & M. Jeckle (Eds.), Web Services, pp. 254--269. Springer, Heidelberg (2004)
13. Li, Q., Liu, A., Liu, H., Lin, B., Huang, L., & Gu, N.: Web Services Provision: Solutions, Challenges and Opportunities (invited paper). In: 3rd International Conference on Ubiquitous Information Management and Communication, pp. 80--87. Suwon, Korea (2009)
14. Martin, D., Burstein, M., McDermott, D., McIlraith, S., Paolucci, M., Sycara, K., et al.: Bringing Semantics to Web Services with OWL-S. World Wide Web. 10, 243—277 (2007)
15. Nayak, R., & Iryadi, W.: XML Schema Clustering with Semantic and Hierarchical Similarity Measures. Knowledge-Based Systems. 20, 336—349 (2007)
16. Nayak, R., & Lee, B.: Web Service Discovery with Additional Semantics and Clustering. In: 9th IEEE/WIC/ACM International Conference on Web Intelligence, pp. 555--558. Fremont, USA (2007)
17. Shafiq, O., Moran, M., Cimpian, E., Mocan, A., Zaremba, M., & Fensel, D.: Investigating Semantic Web Service Execution Environments: A Comparison between WSMX and OWL-S Tools. In: 2nd International Conference on Internet and Web Applications and Services, pp. 31--36. Morne, Mauritius (2007)
18. Van Rijsbergen C.J., Robertson, S.E. and Porter, M.F.: New Models in Probabilistic Information Retrieval. British Library, London (1980)
19. Wu, C., Potdar, V., & Chang, E.: Latent Semantic Analysis - The Dynamics of Semantics Web Services Discovery. In: Advances in Web semantics I: Ontologies, Web Services and Applied Semantic Web, pp. 346--373. Springer, Heidelberg (2009)