The BUAP Participation at the Web Service Discovery Track of INEX 2010*

María Josefa Somodevilla, Beatriz Beltrán, David Pinto,
Darnes Vilariño, and José Cruz Aaron

Faculty of Computer Science
Benemérita Universidad Autónoma de Puebla, México
{mariasg,bbeltran,dpinto,darnes}@cs.buap.mx

Abstract. A first approach for web services discovering based on techniques from Information Retrieval (IR), Natural Language Processing (NLP) and XML Retrieval was developed in order to use texts contained in WSDL files. It calculates the degree of similarity between words and their relative importance to support the task of web services discovering. The first algorithm uses the information contained in the WSDL (Web Service Description Language) specifications and clusters web services based on their similarity. A second approach based on a information retrieval system that index terms by using an inverted index structure was also used. Both algorithms are applied in order to evaluate 25 topics in a set of 1947 real web services (all of them provided by INEX).

1 Introduction

The Service Oriented Architecture (SOA)[1] was developed based on the concept of a wide mesh of collaborating services, published and available for invocation. Web services are the set of protocols by which services are published, discovered, and used in a technology independent, standard form. As the number of web services repositories grows and the number of available services expands, finding the web service that one needs has become a key task within the invocation process. Web service discovery is concerned with locating web services that match a set of functional and non-functional criteria [1]. The Web Services Description Language (WSDL) [2] is the most basic mechanism used to describe web services. This leads many current discovery approaches to focus on locating web services based on their functional description.

An efficient and effective Web services discovery mechanism is important in many computing paradigms including Pervasive Computing, Service-Oriented

* This work has been partially supported by projects: CONACYT #106625, VIEP #SOGJ-ING11-I, #BEMB-ING11-I, as well as by the PROMEP/103.5/09/4213 grant.

[1] http://opengroup.org/projects/soa/doc.tpl?gdid=10632

[2] http://www.w3.org/TR/wsdl.html

S. Geva et al. (Eds.): INEX 2010, LNCS 6932, pp. 347–350, 2011.

Computing, and the most recent Cloud Computing, in which Web services constitute the chief building blocks. The Web Service Discovery track aims to investigate techniques for discovery of Web services based on searching service descriptions provided in Web Services Description Language (WSDL) . Participating groups will contribute to topic development and evaluation, which will then allow them to compare the effectiveness of their XML retrieval techniques for the discovery of Web services. This will lead to the development of a test collection that will allow participating groups to undertake future comparative experiments. The rest of this paper is devoted to explain the two different approaches submitted to the competition, as well as the dataset used in the experiments.

2 Description of the Presented Approaches

Two algorithms based on Clustering and Information Retrieval in order to find the most appropiate web service (WSDL file) to a given topic were developed.

The description of the approach that uses a clustering method follows.

1. Tag removal: A corpus was built with the content of the XML tags for each document, in addition to the attribute values of the labels.
2. Parsing WSDL: Stopwords and punctuation symbols were removed from the corpus.
3. Tokenization: The Maximum Matching Algorithm (MMA) algorithm was applied [2], using a list of 53,000 English words which split them into tokens (i.e. *GetAllitems* as *Get All Items*).
4. Re-parsing WSDL: Stopwords and punctuation symbols were removed from the corpus again due to the MMA decomposition.
5. Word stemming: The Porter stemming algorithm was applied to the corpus.
6. K-means algorithm: K=2 was used; the distance criterion NGD is presented in Eq. (1), and the convergence criterion is that the centroid words are at least twice in different iterations.
7. Content word recognition: Thereafter, we removed the words of the cluster with minimal elements (i.e. *service*, *SOA*, *array* and *data*).
8. Services corpus creation: A second corpus was constructed with the services of each XML file; again we used the MMA algorithm, we eliminate stopwords, and finally we applied the Porter algorithm.
9. Query answering: Using the two corpus constructed, a query can be answered by applying Eq. (2), and then sorting the results from lowest to highest.

$$\mathrm{NGD}(x,y)=\frac{\max\ \{logf(x),logf(y)\}-logf(x,y)}{logM-\min\ \{logf(x),logf(y)\}} \tag{1}$$

$$O(S_i,S_j)=0.5*S'(S_i,S_j)+0.5*S''(S_i,S_j) \tag{2}$$

where:

$$S'(S_i,S_j)=\frac{\sum_{a\in S_i}\sum_{b\in S_j}Sim(a,b)}{|S_i||S_j|} \tag{3}$$

$$S''(S_i, S_j) = 1 - \mathrm{NGD}(S_i, S_j) \tag{4}$$

Following we describe the approach that uses information retrieval for finding the corresponding web services files that satisfies the user needs.

The implementation based on NLP uses an inverted index for storing all the terms detected in the WSDL files. For the case of function names, we have also used the MMA algorithm. Each term is used as the dictionary entry in the data structure, and one posting list is attached to each dictionary entry. Finally, given a query, we may calculate the intersection between pairs of posting lists (p_1 and p_2) as shown in the Algorithm 1 (taken from [3]).

Algorithm 1. Intersection of two posting lists

```
Input: Posting lists $p_1$ and $p_2$
Output: Relevant documents $D_1, D_2, \cdots$
$answer = \langle\rangle$
while $p_1 != NIL$ and $p_2 != NIL$ do
    if $docID(p_1) = docID(p_2)$ then
        $\mathrm{ADD}(answer, docID(p_1))$;
        $p_1 = next(p_1)$;
        $p_2 = next(p_2)$;
    else
        if $docID(p_1) < docID(p_2)$ then
            $p_1 = next(p_1)$
        else
            $p_2 = next(p_2)$
        end
    end
end
return $answer$
```

3 Experimental Results

In Figure 1 we may see the interpolated precision, whereas in Table 1 we show the mean average precision obtained by the teams at the competition. The clustering-based approach obtained a low performance, and, therefore, we will discard the use of this technique in future experiments. With respect to the other approach, we did not obtained an official evaluation due to a format problem with our output file. However, a preliminar evaluation (using our own evaluation tools) show a very good performance. In summary, we consider that the use of techniques of information retrieval obtains the best performance in this kind of task, when the correct features are extracted from wsdl files.

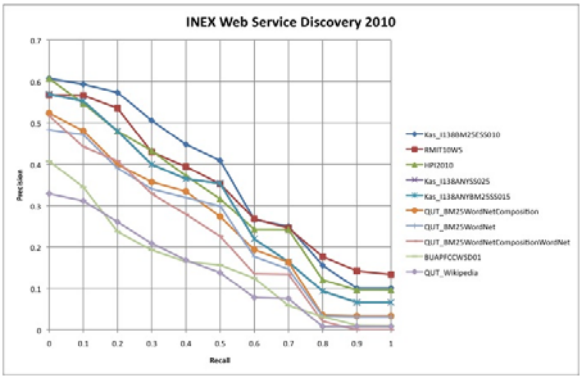

Fig. 1. Interpolated precision at standard recall levels

Table 1. Results at the INEX Web Service Discovery Track

Num	**map**	**Institute**	**Run**
1	0.3469	Kasetsart University	Kas_I138BM25ESS010
2	0.3239	RMIT University	RMIT10WS
3	0.2946	Hasso-Plattner-Institut	HPI2010
4	0.2798	Kasetsart University	Kas_I138ANYSS025
5	0.2798	Kasetsart University	Kas_I138ANYBM25SS015
6	0.2348	Queensland Univ. of Technology	QUT_BM25WordNetComposition
7	0.2233	Queensland Univ. of Technology	QUT_BM25WordNet
8	0.2042	Queensland Univ. of Technology	QUT_BM25WordNetCompWordNet
9	0.1451	B. Univ. Automa de Puebla	BUAPFCCWSD01
10	0.1268	Queensland Univ. of Technology	QUT_Wikipedia
11	0.1095	Queensland Univ. of Technology	QUT_WikipediaComposition
12	0.0937	Queensland Univ. of Technology	QUT_WikipediaCompositionWordNet

4 Conclusions

We have presented details about the implemented approaches for tackling the problem of webservice discovery. Two different approaches were implemented, one based on clustering and the second on information retrieval techniques. In general we may see that the second approach behaves better than the one based on clustering.

References

1. Dan, A., Davis, D., Kearney, R., Keller, A., King, R., Kuebler, D., Ludwig, H., Polan, M., Spreitzer, M., Youssef, A.: Web services on demand: Wsla-driven automated management. IBM Systems Journal 43(1), 136–158 (2004)
2. Guodong, Z.: A chunking strategy towards unknown word detection in chinese word segmentation. In: Dale, R., Wong, K.-F., Su, J., Kwong, O.Y. (eds.) IJCNLP 2005. LNCS (LNAI), vol. 3651, pp. 530–541. Springer, Heidelberg (2005)
3. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2009)