AN IMPROVED APPROACH FOR MEASURING SIMILARITY AMONG SEMANTIC WEB SERVICES

Pedro Bispo Santos, Leandro Krug Wives and José Palazzo Moreira de Oliveira
PPGC, Instituto de Informática, UFRGS, Porto Alegre, Brazil

Keywords: Web Services, Similarity, Matchmaking.

Abstract: The current paper presents an improved approach for an ontology-based semantic Web service similarity assessment algorithm. The algorithm uses information from IOPE (Inputs, Outputs, Preconditions, Effects) categories, searching for information about the concepts located in these categories, analyzing how they are related in an ontology taxonomy. Experiments performed using a data set containing 1083 OWL-S semantic web services show that the improved approach increases the algorithm precision, decreasing the number of false positives in the retrieved results, and still having a good recall. Furthermore, this work presents the parameters that were used to achieve better precision, recall and f-measure.

1 INTRODUCTION

Web Services provide an interesting solution for software applications' interoperability due to its XML-based standards, i.e., SOAP, WSDL, and UDDI. Despite being a standard, UDDI has many issues. One of them is that UDDI does not provide a sophisticated method for querying its registries. Queries usually consist of simple keywords, and require some previous knowledge about the registries, like business entity key number or name. It also does not rank the retrieved results, which can be a huge problem in public registries because of the advent and the continuous growing of the Service Web, the set of web services available on the Web (Yu et al., 2008). Thus, using UDDI for web service's discovery is not an optimal option, since the queries cannot fully express the user's need and the services retrieved are not ranked.

The discovery issue is mainly related to the lack of expressivity offered by WSDL, which is an entirely syntactic description language defining a Web service interface by listing its operations, data types and user-defined types present in the operations' input and outputs, besides binding information. A richer language is necessary for tuning the web service discovery process (Petrie, 2009), and that is the objective of Semantic Web Services (McIlraith et al., 2001). Examples of richer languages are WSMO (Bruijn et al., 2005), OWL-S (Martin et al., 2004), and SAWSDL (Kopecky et al., 2007), the first two being W3C member submissions.

In this paper, we present an approach that improves an existing similarity assessment algorithm (Liu et al., 2009), in order to calculate the similarity between semantic Web services. The current work improves the aforementioned algorithm by changing the way similarities are calculated, resulting in a much better precision and recall, as it is pointed out by our experiments (section 4). Furthermore, this work presents an analysis regarding which configuration of parameters presents the better results, since the previous approach contains different parameters, but the authors did not make any experiment testing which ones would provide a better precision, recall and f-measure.

The remaining of this paper is organized as follows: section 2 reviews the state-of-art in semantic web service similarity algorithms; the algorithm proposed by Liu (Liu et al., 2009) and the improvement made by this work is presented in section 3; section 4 shows the results obtained by the experiments realized; conclusions obtained and further research directions are described in section 5.

2 RELATED WORK

There are several semantic web services matchmakers in the literature. OWLS-MX (Klusch et al., 2009) uses a hybrid approach, using semantic and syntactic information, that is, it uses logic based reasoning and non-logic based information retrieval techniques. The

Bispo Santos P., Krug Wives L. and Palazzo Moreira de Oliveira J..
AN IMPROVED APPROACH FOR MEASURING SIMILARITY AMONG SEMANTIC WEB SERVICES.
DOI: 10.5220/0003894600830088
In *Proceedings of the 8th International Conference on Web Information Systems and Technologies* (WEBIST-2012), pages 83-88
ISBN: 978-989-8565-08-2

information retrieval techniques used by them were cosine, extended Jaccard, information loss, and Jensen-Shannon information divergence. Nevertheless, it does not consider preconditions and effects. This algorithm uses information present in the input and output categories only. The work presented in this paper considers preconditions and effects, and does not take information retrieval techniques into consideration.

The approach used by Wei et al. (Wei et al., 2008) is similar to Klusch et al. (Klusch et al., 2009) since it uses only I/O information, and combines it with syntactic information. In this case, it uses information extraction techniques for generating a constraint graph and then matchmaking the similarity. However, this extraction is made on textual description, and since the user is not obliged to provide textual description, this can be a serious issue for the efficiency of this approach.

Khdour and Fasli (Khdour and Fasli, 2010) proposes a method for filtering the relevant semantic web services for a query, for diminishing the amount of time necessary for calculating the similarity among the relevant semantic web services and the query. However, their work only determine if, a priori, a semantic web service is relevant or not for a query in a binary way, for then using some similarity algorithm for providing a rank among the relevant semantic web services.

Kritikos and Plexousakis (Kritikos and Plexousakis, 2006) points out that syntactic based discovery techniques presents results with low precision and high recall ratios. A richer language is necessary for tuning the web service discovery process (Petrie, 2009), and that is the objective of Semantic Web Services (McIlraith et al., 2001).

This richer language must be both human and computer readable, having good expressivity, wherein such expressivity does not imply in losing decidability, that is, every reasoning made in this language will be finished in a feasible time. The semantic Web idea (Shadbolt et al., 2006) is that software agents can automate most of the tasks done by human agents. Thus, the utilization of these semantic description languages would ease the process of web service discovery for these software agents.

Liu et al.(Liu et al., 2009) present an ontology-based algorithm for measuring the similarity among semantic web services. It is based on Li et al.'s (Li et al., 2003) work, which uses information present in a hierarchical semantic knowledge base of words for calculating the similarity among different words. Liu and partners apply it to calculate similarity among semantic web services by using a domain ontology taxonomy. It uses information present in a web service profile description, which, in fact, contains information about web service's inputs, outputs, preconditions and effects, wherein all these categories are considered as sets of concepts.

Li et al (Li et al., 2011) presents a different kind of similarity measurement, the behavioral web services similarity. It states that there are three kinds of similarity: syntactic, semantic and behavioral. And his work focuses on the latter one, which consists in analyzing how the exchange of messages occurs, forming a colored petri net for each web services and measuring the behavior similarity based on these coloured petri nets.

It is extremely important that these similarity algorithms present a high precision ratio, due to its increasing adoption, e.g. Maamar et al. (Maamar et al., 2011) uses Liu's work for building an initial social network for each web service present in a given registry. Unfortunately, experiments performed show that Liu's algorithm presents low precision and high recall ratios, bringing too many false positives, resembling syntactic based discovery techniques (Kritikos and Plexousakis, 2006). Furthermore, Liu's approach contains different parameters, but they did not make any experiment testing which ones would provide a better precision, recall and f-measure. The current work improves Liu's algorithm by changing the way similarities are calculated, resulting in a much better precision and recall.

3 SIMILARITY ALGORITHM

The algorithm proposed by Liu et al.'s (Liu et al., 2009) is about calculating similarity among semantic web services by analyzing the relationship among concepts given by an ontology taxonomy. This algorithm is based on Li et al.'s (Li et al., 2003) work for calculating similarity among words by using a hierarchical semantic knowledge base of words, which also takes into account the structure (location, hierarchy) of these words in the taxonomy. An example of a hierarchical knowledge base of words is depicted at Figure 1.

An intuitive way of calculating the similarity between two words consists on evaluating the length of the path that is needed to reach one word from another. For instance, considering Figure 1, the word *boy* is more similar to the word *girl* than to the word *teacher*, since the path from *boy* to *girl* is *boy-male-person-female-girl*, and from *boy* to *teacher* is *boy-male-person-adult-professional-educator-teacher*. Nevertheless, this way is not the

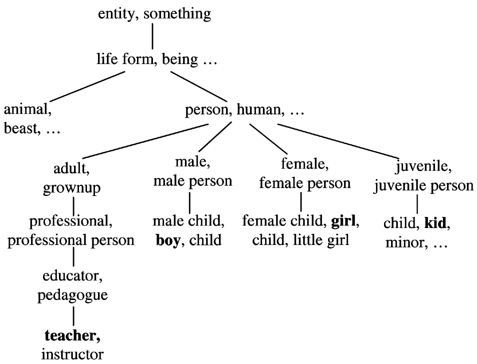

Figure 1: Hierarchical semantic knowledge base of words (Li et al., 2003).

optimal way of measuring the semantic similarity among words because it indicates that *boy* is more similar to *animal* than to *teacher*. Thus, to measure the semantic similarity between two words, Li suggests using the depth of the subsumer word of the two words and their semantic density along with the path length for reaching one word from another. Therefore, given two words w_1 and w_2, their semantic similarity is:

$$s(w_1,w_2) = f(f_1(l), f_2(h), f_3(d)) \quad (1)$$

being l the shortest path from w_1 to w_2, h the depth of the subsumer and d the semantic density of the words. For calculating $f_1(l)$, it is used the following equation:

$$f_1(l) = e^{-\alpha l}$$

being α a smoothing factor. The exponential form ensures that it stays in the [0,1] range (Li et al., 2003). For $f_2(h)$:

$$f_2(h) = \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}$$

being β another smoothing factor, and $\beta > 0$, as $\beta \to \infty$ the depth parameter is not considered in the measurement.

For measuring the semantic density among two words w_1 and w_2 a corpus is needed, since each word has an information gain, and this value is based on the probability of finding this word among others in a given corpus (based on their frequency in the corpus). Thus the semantic density d can be calculated in the following manner:

$$d = max_{c \in sub(w_1,w_2)}[-logp(c)]$$

it is the maximum information gain value among all common subsumers of the two words. Then, for measuring $f_3(d)$:

$$f_3(d) = \frac{e^{\lambda d} - e^{-\lambda d}}{e^{\lambda d} + e^{-\lambda d}}$$

being λ a smoothing factor just as β.

3.1 Semantic Web Service Similarity

The idea of using semantic annotations for describing web services interfaces came from (McIlraith et al., 2001), and its main objective is to allow software agents to automate the discovery, composition and invocation process. The algorithm presented by Liu et al. uses four categories present in a semantic web service profile: Preconditions, Inputs, Outputs, and Effects. Each category is considered as a set of concepts, hence, given two categories, Liu et al. define the similarity among categories as being:

$$D_s(C_1,C_2) = \sum_{c_1 \in C_1, c_2 \in C_2} ws(c_1,c_2) \quad (2)$$

being c_1 a concept from the category C_1, c_2 a concept from the category C_2, and w the weight for the i-th pair of concepts. This equation gives all the possible pairs of concepts among the two categories, and if any similarity, calculated by (1), is equal to zero, then zero is assigned for the pair weight w, being $\sum w = 1$. Then, given two semantic web services, their similarity is measured by the following equation:

$$S = \sum_i W_i D_s(C_{i1}, C_{i2})$$

being W_i the weight for the category pair, wherein $\sum W_i = 1$. The category pair has to be one of the four categories present in a semantic web service profile: Inputs, Outputs, Preconditions and Effects. Unlike the pair of concepts, here the categories only make pair with their own type.

3.2 Algorithm Improvement

The problem with the algorithm proposed by Liu et al. is depicted in Figure 2. Assuming that concepts *A* and *C* are from one category and concepts *B* and *D* are from another category, Liu et al.'s algorithm calculates the similarities among those concepts and then uses all those values as a bag of concepts for assessing the similarity among the categories.

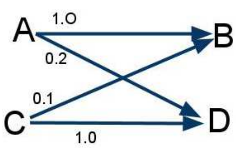

Figure 2: Concept matching.

In the case depicted in Figure 2, clearly the similarity value will be prejudiced because of the low similarity of concept *A* with concept *D* and concept *B* with concept *C*. The similarity value among the categories should be 1 in this case, but it will not be according to Liu et al.'s algorithm. Hence, the improvement proposed here considers only the highest similarity score from each concept with the concepts of the other category instead of using all the similarity scores among every possible pair of concept. In other words, for each concept present in one category, this concept will make a pair with each concept from the other category, each pair will have a similarity degree, and only the highest similarity degree will be considered. Then, instead of using (2) for calculating similarities among categories, this measurement is improved by the following:

$$D_s(C_1,C_2) = \sum_{c_1 \in C_1} w_{c_1} max_{c_2 \in C_2}(s(c_1,c_2))$$

The experiments realized in section 4 show that this improvement results in a much better precision, since Liu et al.'s algorithm brings too many false positives, having high recall ratios and low precision. Although this approach has a smaller recall, the ratio is acceptable, as pointed out by f-measure (see next section).

4 EXPERIMENTS

OWLS-TC4 (OWL-S Test Collection 4) was used for performing the experiments. It is available for download at http://www.semWebcentral.org/projects/owls-tc/. This data set is composed by 1083 services described in OWL-S 1.1 from nine different domains. Part of these web services was obtained by UBRs (Universal Business Records), which are now extinct[1]. There are 42 queries, and an XML document defining the relevant services for each query. A query is described as a service itself, defined by its inputs, outputs, preconditions and effects. Different users subjectively assessed the service request/offer pairs. The pooling strategy used is one similar to the one used at TREC7. The ontologies for describing the concepts present in the Web services inputs and outputs were obtained from public sources on the Web. More details about this data set are available at the manual that comes along with the OWLS-TC4 data set.

The algorithm implementation was done using the Jena API[2] for reading the OWL ontologies, and the OWL-S API[3] for reading the semantic Web services' interfaces. Although there are other data sets available, OWLS-TC was chosen because OWL-S is currently the most used language for describing semantic web services on the Web (Klusch and Zhing, 2008). The preconditions and effects rules are in SWRL and in PDDL. Nevertheless, according to (Klusch et al., 2009), the majority of the accessible OWL-S services do not specify preconditions and effects, so, these two categories were not considered in the implementation for now.

Three information retrieval measures were used:

$$precision = \frac{|\{relevant\_documents\} \cap \{retrieved\_documents\}|}{|\{retrieved\_documents\}|}$$

$$recall = \frac{|\{relevant\_documents\} \cap \{retrieved\_documents\}|}{|\{relevant\_documents\}|}$$

$$f-measure = 2 * \frac{precision * recall}{precision + recall}$$

Implementation A was performed as suggested in this work, and implementation B was conducted accordingly to Liu et al., and they did not present the configuration for which the parameters are the best. They present, however, two relevant configurations for measuring similarities among words: path length and subsumer depth, with $\alpha = 0.2$ and $\beta = 0.6$; considering only the path length with $\alpha = 0.25$.

Table 1 presents the averaged results for the first configuration: considering the path length and the subsumer depth, using $\alpha = 0.2$ and $\beta = 0.6$. Figure 3 shows the exponential function with $\alpha = 0.2$, if the path length is greater than 4, then the similarity value among the concepts will be decreased. Figure 4 shows the monotonic function with $\beta = 0.6$, and if the depth of the subsumer is greater than 2, then the similarity value among the concepts will be increased. The results show that the implementation A has a much better precision than implementation B, 85.26% against 34.96%. Despite the implementation A recall is a little bit worse than implementation B, 69.41% against 71.72%, the f-measure shows that implementation A presents a better result 71.28% against 41.72%.

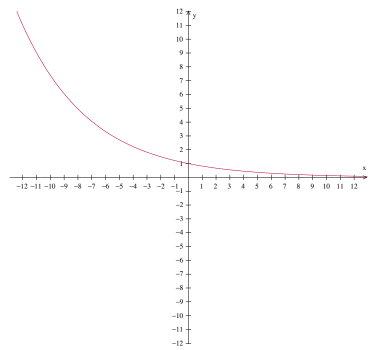

Figure 3: Exponential function, with $\alpha = 0.2$.

Despite the fact that the first configuration presented good results, table 2 shows that the second one did not achieve the same. It only considers the path length in the similarity measurement among two concepts, with $\alpha = 0.25$. Although both implementations

[1] http://soa.sys-con.com/node/164624/

[2] Available at http://jena.sourceforge.net/.

[3] Available at http://on.cs.unibas.ch/owls-api/.

Table 1: Experiments - $\alpha = 0.2, \beta = 0.6$.

	Implementation A			**Implementation B**		
query	**precision**	**recall**	**f-measure**	**precision**	**recall**	**f-measure**
Average	85.26%	69.41%	71.28%	34.96%	71.72%	41.72%

Table 2: Experiments - $\alpha = 0.25$.

	Implementation A			**Implementation B**		
query	**precision**	**recall**	**f-measure**	**precision**	**recall**	**f-measure**
Average	3.60%	91.23%	6.80%	3.12%	81.66%	5.92%

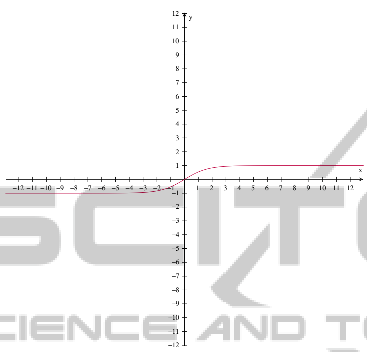

Figure 4: Depth function, with $\beta = 0.6$.

did not present good results, implementation A had a better precision, recall and f-measure.

5 CONCLUSIONS AND FUTURE WORK

This paper presented an improvement for an ontology-based algorithm for performing similarity assessment among semantic web services. This improvement resulted in a more accurate algorithm, as pointed out by the results obtained in the experiments presented in the previous section. As future work, further experiments will be performed using full information from the semantic web services' interfaces, i.e., using preconditions and effects categories. Further, we should also consider the use of other data sets, and different description languages like SAWSDL.

ACKNOWLEDGEMENTS

This research was partially supported by CAPES and CNPq, Brazil.

REFERENCES

Bruijn, J. D., Bussler, C., Domingue, J., Fensel, D., Hepp, M., Keller, U., Kifer, M., Konig-Ries, B., Kopecky, J., Lara, R., Lausen, H., Oren, E., Polleres, A., Roman, D., Scicluna, J., and Stollberg, M. (2005). Web service modeling ontology (wsmo). In *W3C Recommendation*.

Khdour, T. and Fasli, M. (2010). A semantic-based web service registry filtering mechanism. In *Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on*, pages 373 –378.

Klusch, M., Fries, B., and Sycara, K. (2009). Owls-mx: A hybrid semantic web service matchmaker for owl-s services. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(2):121 – 133.

Klusch, M. and Zhing, X. (2008). Deployed semantic services for the common user of the web: A reality check. In *Semantic Computing, 2008 IEEE International Conference on*, pages 347 –353.

Kopecky, J., Vitvar, T., Bournez, C., and Farrell, J. (2007). Sawsdl: Semantic annotations for wsdl and xml schema. *IEEE Internet Computing*, 11(6):60 –67.

Kritikos, K. and Plexousakis, D. (2006). Semantic qos metric matching. In *Web Services, 2006. ECOWS '06. 4th European Conference on*, pages 265 –274.

Li, X., Fan, Y., Sheng, Q., Maamar, Z., and Zhu, H. (2011). A petri net approach to analyzing behavioral compatibility and similarity of web services. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 41(3):510 –521.

Li, Y., Bandar, Z. A., and McLean, D. (2003). An approach for measuring semantic similarity between words using multiple information sources. *IEEE Transactions on Knowledge and Data Engineering*, 15:871–882.

Liu, M., Shen, W., Hao, Q., and Yan, J. (2009). An weighted ontology-based semantic similarity algorithm for web service. *Expert Systems with Applications*, 36(10):12480–12490.

Maamar, Z., Santos, P., Wives, L., Badr, Y., Faci, N., and Palazzo M. de Oliveira, J. (2011). Using social networks for web services discovery. *Internet Computing, IEEE*, 15(4):48 –54.

Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Maryland, Payne, T., Sirin, E., Srinivasan, N., and Sycara, K. (2004). Owl-s: Semantic markup for web services. In *W3C Recommendation*.

McIlraith, S., Son, T., and Zeng, H. (2001). Semantic web services. *IEEE Intelligent Systems*, 16(2):46 – 53.

Petrie, C. (2009). Practical web services. *Internet Computing, IEEE*, 13(6):93 –96.

Shadbolt, N., Hall, W., and Berners-Lee, T. (2006). The semantic web revisited. *Intelligent Systems, IEEE*, 21(3):96 –101.

Wei, D., Wang, T., Wang, J., and Chen, Y. (2008). Extracting semantic constraint from description text for semantic web service discovery. In Sheth, A., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., and Thirunarayan, K., editors, *The Semantic Web - ISWC 2008*, volume 5318 of *Lecture Notes in Computer Science*, pages 146–161. Springer Berlin / Heidelberg.

Yu, Q., Liu, X., Bouguettaya, A., and Medjahed, B. (2008). Deploying and managing web services: issues, solutions, and directions. *The VLDB Journal*, 17:537–572.