Empirical Study for Semantic Annotation of Web Services

Djelloul Bouchiha¹, Mimoun Malki¹, Djihad Djaa², Abdullah Alghamdi³, Khalid Alnafjan³

¹ EEDIS Laboratory, Djillali Liabes University of Sidi Bel Abbes, Algeria
bouchiha.dj@gmail.com, malki@univ-sba.dz

² University Dr. Taher Moulay, Computer Department, Saida, Algeria
djihad22@hotmail.fr

³ College of Computer and Information Sciences. King Saud University, Riyadh, Saudi Arabia
{ghamdi, alnafjan}@ksu.edu.sa

Received 21 December 2012
Accepted 19 November 2013

Abstract

Web services have become the main paradigm for the development of distributed software systems using a common set of technologies, including SOAP, WSDL and UDDI. This allows accessing to software components residing on different platforms and written in different programming languages. However, several tasks, including service discovery and composition, remain difficult to be automated. Thus, a new technology has emerged to solve this problem; it is the Semantic Web Services (SWS). One way to produce SWS is the annotation. In this paper, an approach to annotate Web services is presented. The approach consists of two main processes, categorization and matching. Both processes use ontology matching techniques. In particular, the two processes use similarity measures between entities, strategies to calculate similarities between sets and a threshold corresponding to the accuracy. Thus, an internal comparative study has been done to determine which strategy is appropriate to this approach, which measure gives best results and which threshold is optimum for the selected measure and strategy. An external comparative study has been also carried out to prove the efficacy of this approach compared to existing annotation approaches.

Keywords: Annotation, Web Service, SAWSDL, Semantic Web Services, Ontology Matching.

1. Introduction

A Web service is software which provides its functionality through the Web with common standards, including SOAP, WSDL and UDDI. Due to the syntactic nature of these standards, discovery and composition of these services have become difficult. To solve this problem, semantic Web services have emerged. To add semantics to a service, it is possible to annotate the elements of this service with the concepts of a domain ontology. The annotation consists in associating the WSDL elements of a Web service with the concepts of an existing semantic model. Often this model is a domain ontology of the Web service.

In a previous work, an annotation approach has been proposed. It consists of two main processes: categorization which classifies the WSDL document in its corresponding domain, and matching which associates each entity of the WSDL document with the corresponding entity in the domain ontology. Both categorization and matching are based on ontology

---

¹ http://www.w3.org/TR/soap/
² http://www.w3.org/TR/wsdl
³ http://www.uddi.org/
matching techniques\textsuperscript{2} which in turn use similarity measures between entities. A similarity measure quantifies how much two entities are similar. In particular, WordNet based similarity measures\textsuperscript{3} are used.

To compare the results from the annotation approach using different similarity measures (internal comparison), and compare this approach with other existing approaches (external comparison), a tool called SAWSDL Generator has been implemented. The tool receives as input a WSDL file and a set of domain ontologies, and then generates a WSDL document annotated according to the SAWSDL\textsuperscript{4} standard.

This paper presents an internal comparative study to improve, optimize and determine under which conditions the annotation approach provides its best results. An external comparative study is also presented to show clearly the effectiveness of this approach over other annotation approaches.

The paper is organized as follows: the next section presents a summary of the Web service annotation approaches which exist in the literature. Section 3 explains the annotation approach presented in Ref. 1. Section 4 and Section 5 detail a comparative study. The final section concludes our work while presenting the main contributions that the comparative study enabled us to provide.

2. Literature Review

The annotation of a Web service consists in associating and tagging WSDL elements of this service with the concepts of an ontology.\textsuperscript{5}

Several approaches have been proposed for annotating Web services. Table 1 summarizes the characteristics of the Web service annotation approaches as follow: (1) The "Approach in" column corresponds to the approach reference in question; (2) The "Considered elements" column describes the considered elements in the annotation process; (3) The "Annotation resource" column indicates the model from which semantic annotations are extracted; (4) The "Techniques" column presents the used techniques for the annotation; (5) The "Tool" column indicates the tool supporting the approach.

<table>
<thead>
<tr>
<th>Approach in</th>
<th>Considered elements</th>
<th>Annotation resource</th>
<th>Techniques</th>
<th>Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ref. 6</td>
<td>Operation parameters</td>
<td>Workflow</td>
<td>Parameter compatibility rules</td>
<td>Annotation Editor</td>
</tr>
<tr>
<td>Ref. 7</td>
<td>Operations, message parts and Data.</td>
<td>Domain ontology</td>
<td>Text classification techniques</td>
<td>ASSAM</td>
</tr>
<tr>
<td>Ref. 8</td>
<td>Natural-language query</td>
<td>Domain Ontology</td>
<td>Text mining techniques</td>
<td>Visual OntoBridge (VOB)</td>
</tr>
<tr>
<td>Ref. 9</td>
<td>Data (Inputs and Outputs of services)</td>
<td>Meta-data (WSDL)</td>
<td>Machine learning techniques</td>
<td>Semantic labelling tool</td>
</tr>
<tr>
<td>Ref. 10</td>
<td>Annotation &amp; Query</td>
<td>Workflow</td>
<td>Propagation method</td>
<td>Prolog Implementation</td>
</tr>
<tr>
<td>Ref. 11</td>
<td>Datalog definitions</td>
<td>Source definitions</td>
<td>Inductive logic search</td>
<td>EIDOS</td>
</tr>
<tr>
<td>Ref. 1</td>
<td>Complex types and operations names</td>
<td>Domain ontology</td>
<td>Ontology matching</td>
<td>SAWSDL Generator</td>
</tr>
<tr>
<td>Ref. 5</td>
<td>Data (Inputs and Outputs of services)</td>
<td>Domain ontology</td>
<td>Schema matching techniques</td>
<td>MWSAF tool</td>
</tr>
</tbody>
</table>

There are also many other tools in semantic annotation like CharaParser which is a software application for semantic annotation of morphological descriptions.\textsuperscript{12} Jonquet et al. (See Ref. 13) developed NCBO Annotator, an ontology-based Web service for the annotation of textual biomedical data with biomedical ontology concepts. SemAF (Semantic annotation framework) allows the specification of an annotation language.\textsuperscript{14} Wyner and Peters (See Ref. 15) propose several instances of GATE’s Teamware to support annotation tasks for legal rules, case factors and case decision elements. Liao et al. (See Ref. 16) identify three main components of semantic annotation, propose for it a formal definition and presents a survey of current semantic annotation methods.

In a preliminary work, Bouchiha et al. (See Ref. 1) propose to annotate Web services using ontology matching techniques. In the rest of this document, we usually use the name of the tool "SAWSDL Generator" to reference this approach.
3. SAWSDL Generator

As shown in Fig. 1, the annotation approach consists of two main processes: categorization and matching. Both categorization and matching are based on ontology matching techniques. The goal of ontology matching is to find relationships between entities. Usually, these relations are equivalences discovered through similarity measures computed between these entities. To be accomplished, the ontology matching process uses similarity measures between entities. A similarity measure quantifies how much two entities are similar.

Fig. 1. SAWSDL Generator Architecture (See Ref. 1)

SAWSDL Generator uses similarity measures based on WordNet, such as Path, Resnik, Lin and Jiang. WordNet is an online lexical database designed for use in a program. Thus, these measures are calculated and then normalized. Normalization usually involves reversing the measured value to obtain a new value between 0 and 1. The value 1 indicates a total semantic equivalence between two entities.

When multiple ontologies are available, similarities between sets have to be calculated by comparing the set of entities of the WSDL file with all entities of each ontology. Based on these similarities, the system chooses one among ontologies to run the matching algorithm. The selected domain ontology determines the category of the WSDL document. This process is called "Categorization Process".

SAWSDL Generator treats an ontology as a set of entities (concepts), and a WSDL document also as a set of entities (XSD data types, interfaces, operations and messages). In data analysis, the linkage aggregation methods, including Single Linkage, Full Linkage and Average Linkage, allow the assessment of the distance between two sets of elements.

After the categorization process, the elements of the WSDL document are associated to the ontology concepts using a similarity measure. This process is called "Matching Process".

The whole annotation process is a semi-automatic process where the user can intervene to validate, modify or reject the results. He can also set a threshold for the similarity measure. The threshold represents the required accuracy; it is a value between 0 and 1; the value 1 indicates that there must be a total semantic equivalence between two concepts. The use of a threshold reflects the allowable tolerance during calculations. More the threshold value is larger, more the results are accurate.

In the next section, a comparative study allows to evaluate and compare the results of the annotation process using different similarity measures.

4. Internal Comparison

The purpose of the internal comparison is to determine the measure, the strategy and the threshold that return the best results for the annotation process.

SAWSDL Generator offers the user the ability to choose the measure, the strategy and the threshold used in the categorization and matching. Thus, the comparative study is carried out for each one of two processes:

4.1. Categorization evaluation

The categorization allows to associate a service to its corresponding domain. In this section, we use the term categorization or classification to say the same thing.

To check the categorization process with different similarity measures, we used a corpus of 424 Web services organized manually into 26 areas (categories). Because of the lack of domain ontologies we limited our study to 40 services organized as follows: 14 business

---

domain, 2 travel, 4 money, 8 weather, 6 web, 5 mathematics and one service from the music domain. The domain ontologies used in the categorization process are business** and travel††. As strategy for calculating similarity between sets we have chosen the Average Linkage where the similarity between two sets of entities is equal to the average similarity of all pairs of entities in both sets.

To evaluate and compare the results of the categorization process for different similarity measures, we use in the following the metrics precision, recall and F-measure.18

- Recall (R): proportion of the correctly classified (categorized) services of all the services of the two considered domains, namely business and travel: R = CEN/CN.
- Precision (R): proportion of the correctly classified (categorized) services of the automatically classified services: P = CEN/EN.

The metrics recall and precision can be combined into a single measure, called F-measure, defined as follows: F-measure = (2 * recall * precision) / (recall + precision).19

The following figure shows the curves of the F-measure of different similarity measures according to the threshold values:

Fig. 2. The F-measure curves for the categorization process with different measures

The graph above shows that for a zero threshold, the different measures give the same results of categorization. However, the difference is between the scope of the different curves. This can be justified by the fact that certain measures return high average compared to others. For example, the Jiang measure gives higher averages compared to the Path and Resnik measures.

With more accuracy (threshold>0) the results of the categorization differ between these measures. The measure Path gives the lowest results of categorization compared to the other measures. We can see that with a threshold between 0.00 and 0.015, the Lin measure is the most effective for the categorization. For this interval, the measures Resnik and Pirro provide best categorizations compared to the Jiang measure. This may be justified by the fact that with the Jiang method several services out of domain are incorrectly classified; which is not the case with the other measures. However, for a threshold in the interval 0.015-0.02, the curves level changes so that the Jiang measure becomes the most powerful. When the threshold increases the Jiang measure classifies correctly most of the services; which is not the case for the measures Resnik, Lin and Pirro. For the same reasons, with a threshold between 0.00 and 0.015, the Pirro measure gives poor results compared to the Lin measure. With a threshold>0.015 the Pirro measure becomes better than the Lin measure but remains low compared to the Jiang measure.

We can deduce that if we use the measures Lin, Resnik and Pirro (usually measures that return low semantic averages), then the threshold must be between 0.00 and 0.015, to have best results. However, if we use the Jiang method, we must increase the threshold.

There is a dependency between the choice of the threshold and the choice of the measure. Note that for a threshold<0.015, it is recommended to use the Lin measure. However, with a threshold>0.015 it is recommended to use the Jiang measure. From the graph, we have also seen that the best classification has been obtained with the Lin measure, and with threshold=0.01.

The following table shows a ranking of measures based on the threshold depending on the quality of the results obtained by each measure:

Table 2. Ranking of the similarity measures with different thresholds

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Lin</th>
<th>Resnik</th>
<th>Jiang</th>
<th>Pirro</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00-0.01</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.01-0.015</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.015-0.02</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.02-0.06</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

Note that for a threshold<0.015, it is recommended to use the Lin measure. However, with a threshold>0.015 it is recommended to use the Jiang measure. From the graph, we have also seen that the best classification has been obtained with the Lin measure, and with threshold=0.01.

Published by Atlantis Press
Copyright: the authors

38
A measure of lower rank is better compared to a measure of higher rank for the considered interval of threshold.

4.2. Matching evaluation

We define the function Match which associates a WSDL element with the closest concept in the domain ontology. The function Match is defined as follows:

\[ \text{Match} : E_{\text{wsdl}} \to E_{\text{ontology}} \]
\[ e \rightarrow c, \text{ such that } \sim(e, c) = \max_{c'} \sim(e, c') \]
\[ \text{and } \sim(e, c) \geq \text{threshold} \]

With \( E_{\text{wsdl}} \) the set of elements of the WSDL document, \( E_{\text{ontology}} \) the set of concepts of the domain ontology, \( e \) is an element of \( E_{\text{wsdl}} \), \( c \) is a concept of \( E_{\text{ontology}} \), \( n \) the number of concepts of the domain ontology and \( \sim \) a similarity measure.

To evaluate the matching process, we tested the performance of the association function Match with different similarity measures. For this, we have chosen a reference WSDL document, annotated manually, and consider it as valid. The selected WSDL document was "TrackingAll". As a similarity strategy we have chosen Average Linkage, and as similarity measures we compared the measures Path, Lin, Resnik, Jiang and Pirro. To evaluate the results of the matching process, we used the metrics precision, recall and F-measure.

Table 3 shows the manual annotations proposed to associate the elements of the WSDL document "TrackingAll" with the concepts of the business ontology.

Table 3. Manual associations

<table>
<thead>
<tr>
<th>Element of the WSDL document</th>
<th>Corresponding concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>set CustomerPermission</td>
<td>Agreement</td>
</tr>
<tr>
<td>set CustomerPermissionResponse</td>
<td>Agreement</td>
</tr>
<tr>
<td>get CustomerTracking</td>
<td>Transition</td>
</tr>
<tr>
<td>get CustomerTrackingResponse</td>
<td>Transition</td>
</tr>
<tr>
<td>get OperatorPermission</td>
<td>Agreement</td>
</tr>
<tr>
<td>get OperatorPermissionResponse</td>
<td>Agreement</td>
</tr>
<tr>
<td>get OperatorsList</td>
<td>Agent</td>
</tr>
<tr>
<td>get OperatorsListResponse</td>
<td>Agent</td>
</tr>
<tr>
<td>set CompanyPermission</td>
<td>Agreement</td>
</tr>
<tr>
<td>get CompanyList</td>
<td>Association</td>
</tr>
<tr>
<td>get CompanyListResponse</td>
<td>Association</td>
</tr>
</tbody>
</table>

The following figure shows the curves of the F-measure of different similarity measures according to the threshold values:

From this graph, we note that with a threshold belonging to 0.05-0.20, the measure Path performs best compared to the other measures.

The results of the measures Jiang and Pirro are low compared to the other measures, despite they return the highest similarities. This can be justified by the fact that with the measures Jiang and Pirro, all the elements containing the term "Tracking" are not correctly associated to their corresponding concepts, unlike Path, Resnik and Lin.

We note also that the curves of Lin and Resnik are similar in that they return the same results for a threshold belonging to 0.00-0.15. For a threshold belonging to 0.15-0.35, Resnik is more efficient compared to Lin. For a threshold > 0.35, Resnik returns no association.
It is therefore clear that the measures Jiang and Pirro return the lowest results for the matching process. With the measures Lin, Path and Resnik, the matching performance depends on the chosen threshold. It should be a compromise between the chosen threshold and the used similarity measure. With a threshold of 0.05 - 0.15, it is recommended to use the measure Path; the measures Resnik and Lin are ranked second.

4.3. Discussion

By analyzing the results of the categorization and matching in the same time, we found that for a given threshold, the quality of results is different for both processes. For example, the best results of matching are obtained with a threshold equal to 0.35. However, with this threshold, services are not classified in any category, i.e. the results of the categorization process are low for a threshold equal to 0.35. If using an appropriate threshold for the categorization process, then the quality of matching drops. This is justified by the fact that the similarity values used in the matching process are high compared to the semantic averages returned when using the strategy Average Linkage to accomplish the categorization process.

This problem is therefore due to the strategy Average Linkage used to calculate the semantic average between a WSDL document and an ontology. For example, the element "company" has a strong relationship with "association", but has no similarity with the other concepts of the business ontology: with the measure Path, 

\[ \text{sim}(\text{company}, \text{association}) = 0.25. \]

With the other concepts, 

\[ \text{sim}(\text{company}, \text{concept } x) = 0. \]

The semantic average between the WSDL document and the ontology = \(0.25/33 = 0.0075\). It is therefore clear that the semantic average (0.0075) used for the categorization is too low compared to the similarity (0.25) used for the matching.

To solve this problem, we have three solutions:

- Choose two different thresholds for the categorization and the matching.
- Use two different strategies for both processes.
- Define a new strategy that allows, with a single threshold, to find some compatibility between the categorization process and the matching process.

We chose the third option, and we proposed a hybrid strategy between Single Linkage and Average Linkage.

The new strategy is defined as follows:

**Definition:** Given a similarity function \(\sigma: O \times O \rightarrow R\). The measure between two sets is a similarity function \(\Delta : 2^O \times 2^O \rightarrow R\), such that:

\[ \forall x, y \subseteq O, \Delta(x, y) = \frac{\sum_{e \in x} \max_{e_j \in y} \sigma(e, e_j)}{|x|}, \text{ with } (e_i, e_j) \in x \times y \]

The following graph illustrates the results of the F-measure of the categorization and the matching process with the Lin measure and the new strategy:

![Fig. 4. The F-measure curves for the categorization and matching with the Lin measure](image_url)

Finally, we can say that with the Lin measure, the new strategy and a threshold = 0.10, the annotation approach works best for the categorization and matching at the same time.

5. External Comparison

This section presents a comparative study between the approach presented in Ref. 1 and the other Web service annotation approaches which exist in the literature. Initially, our intention was to compare the approach presented in Ref. 1 with all other approaches. However, several problems have prevented us from making a complete study:

- Unavailability of technical details to implement these approaches.
- It is difficult to exploit the tools associated with these approaches.

Thus, we limited our study to the MWSAF approach. MWSAF (METEOR-S Web Service Annotation Framework) is a Framework for semi-automatically annotate WSDL descriptions of Web services with relevant ontologies.
5.1. MWSAF

The MWSAF framework allows to compare the WSDL document with each ontology using pattern matching techniques to select the corresponding domain ontology to the WSDL document from a set of ontologies. Then annotate the elements of this document with the concepts of the selected ontology.

MWSAF proposes to convert the two models (WSDL and ontology) to a common representation called SchemaGraph to better accomplish the matching. A SchemaGraph is a set of nodes connected by arcs.

MWSAF uses a measure called match score (MS) to compare two concepts of WSDL SchemaGraph and the ontology SchemaGraph. The MS is calculated as the weighted average of two different measures: element level match (ElemMatch) and schema level match (SchemaMatch).

The following formula is used for calculating the match score (MS):

\[
MS = \frac{w1 \times \text{ElemMatch} + w2 \times \text{SchemaMatch}}{w1 + w2}
\]

Where \(0 \leq w1 \leq 1\) \((0 \leq w2 \leq 1)\)

The weights \(w1\) and \(w2\) indicate respectively the contribution of \(\text{ElemMatch}\) and \(\text{SchemaMatch}\) in the total match score.

\(\text{ElemMatch}\) is the measure of linguistic similarity between two concepts based on their names. The function \(\text{ElemMatch}\) uses various matching algorithms as \(\text{NGram, CheckSynonym, CheckAbbreviations, TokenMatcher, Porter Stemmer, etc.}\)

\(\text{SchemaMatch}\) is the measure of structural similarity between two concepts. The matching of two concepts is based on the match of the tree of sub-concepts.

Using the match score (MS), two other measures are also calculated: \(\text{avgConceptMatch}\) and \(\text{avgServiceMatch}\):

- \(\text{avgConceptMatch}\) indicates the degree of similarity between the matched concepts of the WSDL schema and ontology. This measure is used to decide whether the computed mappings should be accepted for the annotation.
- \(\text{avgServiceMatch}\) helps to categorize the service. It represents the average match of all concepts of a WSDL schema and a domain ontology. The domain ontology with which the best average has been calculated represents the category of service.

The following formulas are used to calculate \(\text{avgConceptMatch}\) and \(\text{avgServiceMatch}\):

\[
\text{avgConceptMatch} = \frac{\sum_{i=1}^{k} MS(m_i)}{k}
\]

\[
\text{avgServiceMatch} = \frac{\sum_{i=1}^{k} MS(m_i)}{n}
\]

Where \(k = \text{no of mapped concepts, } n = \text{no of all concepts}\)

5.2. SAWSDL Generator vs. MWSAF

Purpose:
The overall purpose of the two approaches is to semi-automatically annotate Web services with relevant ontologies. They consist of two phases: the categorization and matching. The categorization allows to classify a WSDL document in its corresponding domain. The matching allows to associate the elements of the WSDL document with their corresponding concepts in the ontology.

Used techniques:
The approach MWSAF uses schema matching techniques. It uses the XML schema of the WSDL document and the schema of the ontology to compare them. SAWSDL Generator uses ontology matching techniques by decomposing the WSDL document into its basic elements (XSD data types, interfaces, operations and messages) and the ontology into its concepts, then compares the two sets.

In both approaches, a similarity (or a semantic average) between the WSDL description and the ontology based on a similarity measure between two concepts is calculated to identify the domain ontology which will be retained for the annotation. Both approaches use a common strategy for the calculation of similarity between sets:

\[
\text{Moyenne} = \frac{\sum_{i=1}^{n} \text{MAX}_{j} \text{sim}(e_i, e_j)}{n}
\]

Where \(n = \text{no of WSDL concepts, } m = \text{no of ontology concepts}\)

Two main differences can be distinguished:

- \(\text{The considered WSDL elements and ontology concepts}\): To accomplish the annotation process, the two approaches compare the WSDL elements with the ontology concepts. MWSAF consider on
the one hand the WSDL elements complexType, elementType, Enumeration and element, and on the other hand all the ontology concepts, namely class, sub-class, instance and attribute. SAWSDL Generator uses the WSDL elements complexType, elementType, element, operation, message and part, and it considers the class and sub-class as ontological concepts.

- The similarity between two concepts: While SAWSDL Generator uses linguistic similarity measures to compare two entities, MWSAF uses MS measure (match score) that is a combination between linguistic similarity and structural similarity where the similarity between two entities may depend on other concepts.

Since both approaches involve two processes, namely the categorization and matching, the comparison will focus on these two processes.

5.3. Comparison at the categorization level

To test the categorization process of the two approaches we have taken the same test base used for the internal comparison: 40 services including 16 of business domain, two of travel, 4 money, 8 weather, 6 web, 5 mathematics and 1 service of music. The domain ontologies selected to apply the categorization process are business and travel. To evaluate the results of the categorization process we used the metrics precision, recall and F-measure.

Fig. 5. The F-measure curves for the categorization process of the two approaches

From Fig. 5, we note that for a threshold>=0.25, both approaches give the same results of categorization. But with a threshold<0.25 the MWSAF approach gives results less efficient compared to SAWSDL Generator. The best results of SAWSDL Generator are obtained with a threshold=0.1.

5.4. Comparison at the matching level

To compare the matching results of SAWSDL Generator and MWSAF approach, we tested the performance of the association function according to the used similarity measure. Recall that the main difference between the two approaches is that SAWSDL Generator uses a linguistic similarity measure based on WordNet, and the MWSAF approach uses the measure MS (match score) which is composed of a linguistic similarity measure and structural similarity between two concepts, which relies on the similarity of sub-concepts.

For testing, we chose a reference annotated WSDL document and considered it as valid. The chosen WSDL document was "TrackingAll". As similarity measure, we chose the Lin measure for SAWSDL Generator, and for the MWSAF approach we used the linguistic similarity measure Ngram. To evaluate the results of the matching process, we used the metrics recall, precision and F-measure.

The following table describes the proposed manual annotations.

<table>
<thead>
<tr>
<th>Element of the WSDL document</th>
<th>Corresponding concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>set_Customer_Permission</td>
<td>Agreement</td>
</tr>
<tr>
<td>set_Customer_PermissionResponse</td>
<td>Agreement</td>
</tr>
<tr>
<td>set_Customer_PermissionResult</td>
<td>Agreement</td>
</tr>
<tr>
<td>get_Customer_Tracking</td>
<td>Transition</td>
</tr>
<tr>
<td>get_Customer_TrackingResponse</td>
<td>Transition</td>
</tr>
<tr>
<td>get_Customer_TrackingResult</td>
<td>Transition</td>
</tr>
<tr>
<td>OPER_Active</td>
<td>Agent, Activity</td>
</tr>
<tr>
<td>set_Operator_Permission</td>
<td>Agreement, Association</td>
</tr>
<tr>
<td>set_Operator_PermissionResponse</td>
<td>Agreement, Association</td>
</tr>
<tr>
<td>set_Operator_PermissionResult</td>
<td>Agreement, Association</td>
</tr>
<tr>
<td>get_Operators_List</td>
<td>Agent</td>
</tr>
<tr>
<td>get_Operators_ListResponse</td>
<td>Agent</td>
</tr>
<tr>
<td>get_Operators_ListResult</td>
<td>Agent</td>
</tr>
<tr>
<td>COMPANY_Active</td>
<td>Association, Activity</td>
</tr>
<tr>
<td>set_Company_Permission</td>
<td>Agreement</td>
</tr>
<tr>
<td>set_Company_PermissionResponse</td>
<td>Agreement</td>
</tr>
<tr>
<td>set_Company_PermissionResult</td>
<td>Agreement</td>
</tr>
<tr>
<td>get_Company_List</td>
<td>Association</td>
</tr>
<tr>
<td>get_Company_ListResult</td>
<td>Association</td>
</tr>
<tr>
<td>getSpecified_Company</td>
<td>Association</td>
</tr>
<tr>
<td>getSpecified_Company_R</td>
<td>Association</td>
</tr>
<tr>
<td>getSpecified_Company_Result</td>
<td>Association</td>
</tr>
<tr>
<td>getSpecified_Tracking_AccessResponse</td>
<td>Transition</td>
</tr>
<tr>
<td>getSpecified_Tracking_AccessResult</td>
<td>Transition</td>
</tr>
<tr>
<td>showAll_Tracking</td>
<td>Transition</td>
</tr>
<tr>
<td>showAll_Tracking_Response</td>
<td>Transition</td>
</tr>
<tr>
<td>showAll_Tracking_Result</td>
<td>Transition</td>
</tr>
</tbody>
</table>
Empirical Study for Semantic Annotation of Web Services

The following figure shows the curves of the F-measure of the two approaches according to the threshold values:

![Graph showing F-measure curves for the matching process of the two approaches](image)

Fig. 6. The F-measure curves for the matching process of the two approaches

From this graph we see that the results obtained by MWSAF are underperforming compared to SAWSDL Generator whatever the chosen threshold.

6. Conclusion and perspectives

To take advantage of Web service technology, an approach has been proposed to annotate syntactic WSDL descriptions of Web services by ontological models (See Ref. 1). The annotation approach has two main processes: categorization and matching. In the first process, the WSDL description of a service is assigned to its corresponding domain. In the second process, the WSDL entities are associated with pre-existing domain ontology concepts. Both processes categorization and matching use similarity measures based on WordNet. A tool, called SAWSDL Generator has been developed to implement the proposed approach.

A comparative study has been conducted to improve the annotation approach and show its effectiveness compared to other works. The evaluation experiments showed that this approach works best with the Lin measure and a threshold equal to 0.1. To create a compromise between the categorization and matching, a new strategy for calculating similarity between sets has been proposed.

SAWSDL Generator, as it stands, provides very satisfactory and encouraging results. However, it still suffers from several problems including the problem of compound words and abbreviated words used as identifiers in the descriptions of services. While these are significant identifiers for service developers, they can not be handled by the Java WordNet API‡‡. Hence the need for linguistic analysis for the separation between the composed parts in a word and identify the important parts (the parts that are meaningful to the domain), and the need to analyze the abbreviations. Until these problems are solved, human intervention remains necessary to correct the incorrect matches.

References

9. K. Lerman, A. Plangprasopchok and C. A. Knoblock, Automatically labeling the inputs and outputs of web services, in Proceedings of the National Conference on

‡‡ http://jwordnet.sourceforge.net/handbook.html


