

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,¹ 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

^{*} <http://www.w3.org/TR/soap/>

[†] <http://www.w3.org/TR/wsdl>

[‡] <http://www.uddi.org/>

matching techniques² which in turn use similarity measures between entities. A similarity measure quantifies how much two entities are similar. In particular, WordNet based similarity measures³ 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 SAWSDL⁴ 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 WDSL elements of this service with the concepts of an ontology.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 1. Summary of the Web service annotation approaches

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

There are also many other tools in semantic annotation like CharaParser which is a software application for semantic annotation of morphological descriptions.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.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.² 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.³

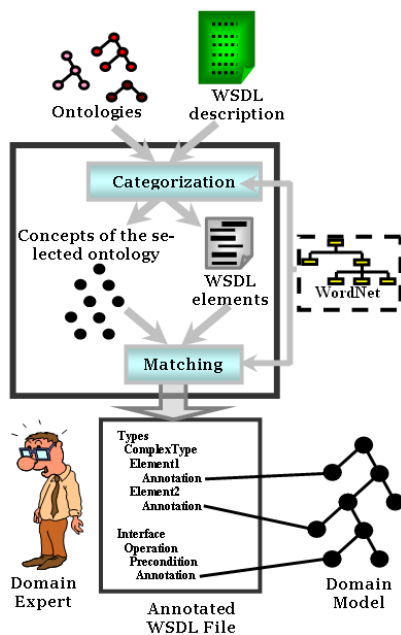


Fig. 1. SAWSDL Generator Architecture (See Ref. 1)

SAWSDL Generator uses similarity measures based on WordNet,³ such as *Path*, *Resnik*, *Lin* and *Jiang*. WordNet¹⁷ 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.²

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).⁷ Because of the lack of domain ontologies we limited our study to 40 services organized as follows: 14 *business*

[§] <http://www.andreas-hess.info/projects/annotator/ws2003.html>

domain, 2 *travel*, 4 *money*, 8 *weather*, 6 *web*, 5 *matematics* 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 (P): 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: $F\text{-measure} = (2 * recall * precision) / (recall + precision)$.¹⁹

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

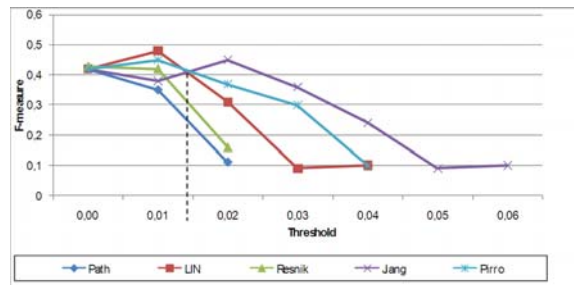


Fig. 2. The F-measure curves for the categorization process with different measures

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

	0.00 - 0.01	0.01 - 0.015	0.015 - 0.02	0.02 - 0.06
1	Lin	Lin	Jiang	Jiang
2	Pirro	Pirro	Pirro	Pirro
3	Resnik	Jiang	Lin	Lin
4	Jiang	Resnik	Resnik	Resnik
5	Path	Path	Path	Path

** <http://www.getopt.org/ecimf/contrib/onto/REA/index.html>

†† <http://protege.cim3.net/file/pub/ontologies/travel/travel.owl>

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:

$$Match: E_{wSDL} \rightarrow E_{ontology}$$

$$e \rightarrow c, \text{ such that } sim(e, c) = MAX_{i=1}^n sim(e, c_i)$$

$$\text{and } sim(e, c_i) \geq threshold$$

With E_{wSDL} the set of elements of the WSDL document, $E_{ontology}$ the set of concepts of the domain ontology, e is an element of E_{wSDL} , c is a concept of $E_{ontology}$, 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

Element of the WSDL document	Corresponding concept
set_Customer_Permission	Agreement
set_Customer_PermissionResponse	Agreement
set_Customer_PermissionResult	Agreement
get_Customer_Tracking	Transition
get_Customer_TrackingResponse	Transition
get_Customer_TrackingResult	Transition
OPER_Active	Agent
set_Operator_Permission	Agreement
set_Operator_PermissionResponse	Agreement
set_Operator_PermissionResult	Agreement
get_Operators_List	Agent
get_Operators_ListResponse	Agent
get_Operators_ListResult	Agent
COMPANY_Active	Association
set_Company_Permission	Agreement
set_Company_PermissionResponse	Agreement
set_Company_PermissionResult	Agreement
get_Company_List	Association
get_Company_ListResult	Association
get_Company_ListResponse	Association

getSpecified_Company	Association
getSpecified_Company_Response	Association
getSpecified_Company_Result	Association
getSpecified_Tracking_Access	Transition
getSpecified_Tracking_AccessResponse	Transition
getSpecified_Tracking_AccessResult	Transition
showAll_Tracking	Transition
showAll_Tracking_Response	Transition
showAll_Tracking_Result	Transition
showSpecified_Tracking	Transition
showSpecified_Tracking_Response	Transition
showSpecified_Tracking_Result	Transition
getSpecified_Tracking	Transition
getSpecified_Tracking_Response	Transition
getSpecified_Tracking_Result	Transition
Show_Tracking_History	Transition
Show_Tracking_HistoryResponse	Transition
Show_TrackingHistoryResult	Transition
get_Tracking_History	Transition
get_Tracking_HistoryResponse	Transition
get_Tracking_HistoryResult	Transition

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

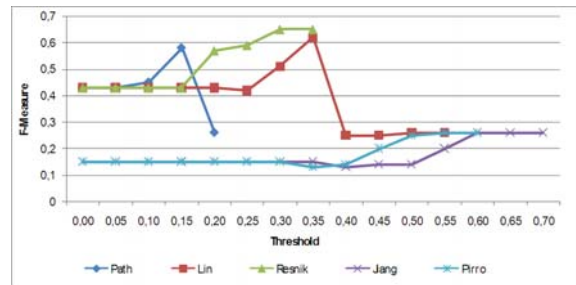


Fig. 3. The F-measure curves for the matching process with different measures

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*, $sim(company, association) = 0.25$. With the other concepts, $sim(company, concept\ x) = 0$. 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 $\Delta: 2^O \times 2^O \rightarrow \mathfrak{R}$, such that:

$$\forall x, y \subseteq O, \Delta(x, y) = \frac{\sum_{i=1}^{|x|} \text{MAX}_{j=1}^{|y|} \sigma(e_i, e_j)}{|x|}, \text{with } (e_i, e_j) \in x * y$$

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:

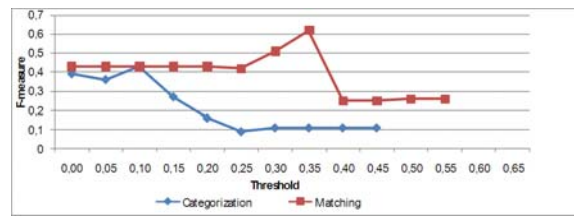


Fig. 4. The F-measure curves for the categorization and matching with the *Lin* measure

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 MWSAF⁵ 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):

$$MS = \frac{w1 * ElemMatch + w2 * SchamaMatch}{w1 + w2}$$

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

The weights $w1$ and $w2$ indicate respectively the contribution of *ElemMatch* and *SchemaMatch* in the total match score.

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

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: *avgConceptMatch* and *avgServiceMatch*:

- *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.
- *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 *avgConceptMatch* and *avgServiceMatch*:

$$avgConceptMatch = \frac{\sum_{i=1}^k MS(m_i)}{k}$$

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

Where $k = no\ of\ mapped\ concepts$, $n = no\ of\ all\ concepts$

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 Generator¹ 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:

$$Moyenne = \sum_{i=1}^n MAX_{j=1}^m sim(e_i, e_j)$$

Where $n = no\ of\ WSDL\ concepts$, $m = no\ of\ ontology\ concepts$

Two main differences can be distinguished:

- *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 *matematics* 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.

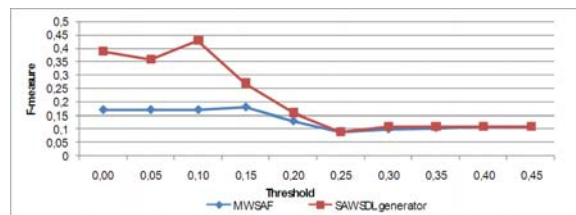


Fig. 5. The F-measure curves for the categorization process of the two approaches

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 4. The proposed manual associations

Element of the WSDL document	Corresponding concept
set_Customer_Permission	Agreement
set_Customer_PermissionResponse	Agreement
set_Customer_PermissionResult	Agreement
get_Customer_Tracking	Transition
get_Customer_TrackingResponse	Transition
get_Customer_TrackingResult	Transition
OPER_Active	Agent, Activity
set_Operator_Permission	Agreement, Association
set_Operator_PermissionResponse	Agreement, Association
set_Operator_PermissionResult	Agreement, Association
get_Operators_List	Agent
get_Operators_ListResponse	Agent
get_Operators_ListResult	Agent
COMPANY_Active	Association, Activity
set_Company_Permission	Agreement
set_Company_PermissionResponse	Agreement
set_Company_PermissionResult	Agreement
get_Company_List	Association
get_Company_ListResult	Association
get_Company_ListResponse	Association
getSpecified_Company	Association
getSpecified_Company_Response	Association
getSpecified_Company_Result	Association
getSpecified_Tracking_Access	Transition
getSpecified_Tracking_AccessResponse	Transition
getSpecified_Tracking_AccessResult	Transition
showAll_Tracking	Transition
showAll_Tracking_Response	Transition
showAll_Tracking_Result	Transition

showSpecified_Tracking	Transition
showSpecified_Tracking_Response	Transition
showSpecified_Tracking_Result	Transition
getSpecified_Tracking	Transition
getSpecified_Tracking_Response	Transition
getSpecified_Tracking_Result	Transition
Show_Tracking_History	Transition
Show_Tracking_HistoryResponse	Transition
Show_Tracking_HistoryResult	Transition
get_Tracking_History	Transition
get_Tracking_HistoryResponse	Transition
get_Tracking_HistoryResult	Transition

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

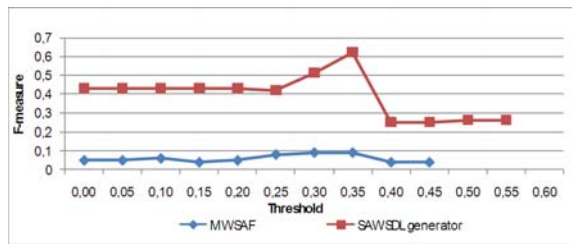


Fig. 6. The F-measure curves for the matching process of the two approaches

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

1. Dj. Bouchiha and M. Malki, Semantic Annotation of Web Services, in the 4th International conference on Web and Information Technologies (ICWIT 2012), (Sidi Bel Abbes, Algeria. April 29-30, 2012).
2. J. Euzenat and P. Shvaiko, *Ontology Matching*, English book, (Springer-Verlag Berlin Heidelberg, 2007).
3. T. Pedersen, S. Patwardhan and J. Michelizzi, WordNet::Similarity - Measuring the Relatedness of Concepts, in *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-2004)*, pp. 1024-1025.
4. J. Farrell and H. Lausen, Semantic Annotations for WSDL and XML Schema, *W3C Recommendation*, (28 August 2007), Available at <http://www.w3.org/TR/sawSDL/>.
5. A. Patil, S. Oundhakar, A. Sheth and K. Verma, METEOR-S Web Service Annotation Framework, *ACM Press*, (WWW 2004), pp. 553-562.
6. K. Belhajjame, S. M. Embury, N. W. Paton, R. Stevens and C. A. Goble, Automatic annotation of web services based on workflow definitions, *ACM Transactions on the Web (TWEB journal)*, 2(2) (2008).
7. A. Hess, E. Johnston and N. Kushmerick, ASSAM: A Tool for Semi-Automatically Annotating Semantic Web Services, *International Semantic Web Conference*, (Hiroshima, Japan, 2004), pp. 320-335.
8. M. Grcar and D. Mladenic, Visual OntoBridge: Semi-automatic Semantic Annotation Software, in *ECML PKDD 2009*, (Bled, Slovenia, September 7-11, 2009), *Proceedings, Part II. LNAI 5782*, (Springer-Verlag Berlin, Heidelberg, 2009), pp. 726-729.
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>

- Artificial Intelligence (AAAI-2006)*. (Boston, Massachusetts, USA. July 2006).
10. S. Bowers and B. Ludäscher, A calculus for propagating semantic annotations through scientific workflow queries. *Query Languages and Query Processing workshop (QLQP-2006)* anised in conjunction with the *10th International Conference on Extending abase Technology*, (2006), pp. 712-723.
 11. M. J. Carman and C. A. Knoblock, Learning Semantic Definitions of Online Information Sources, *Journal of Artificial Intelligence Research*. 30 (2007), pp. 1-50.
 12. H. Cui, CharaParser for fine-grained semantic annotation of organism morphological descriptions, *Journal of the American Society for Information Science and Technology*. 63(4) (April 2012), pp. 738-754.
 13. C. Jonquet, N. Shah, C. Youn, M. Musen, C. Callendar and M. Storey, NCBO Annotator: Semantic Annotation of Biomedical Data, in the *8th International Semantic Web Conference (ISWC 2009) Posters and Demonstrations*, (Washington DC, USA, October 25-29 2009).
 14. J. Pustejovsky, K. Lee, H. Bunt and L. Romary, ISO-TimeML: An International Standard for Semantic Annotation, in *Proc. LREC 2010*, (La Valette, Malte, 2010).
 15. A. Wyner and W. Peters, Semantic Annotations for Legal Text Processing using GATE Teamware, in *the 4th Workshop on Semantic Processing of Legal Texts (SPLeT 2012)*, (Istanbul, Turkey, 27 May 2012).
 16. Y. Liao, M. Lezoche, H. Panetto and N. Boudjlida, Semantic Annotation Model Definition for Systems Interoperability, in *the 6th International Workshop on Enterprise Integration, Interoperability and Networking (EI2N)*, (Hersonissos, Crete : Greece, October 2011).
 17. G. A. Miller, WordNet: An on-line lexical database, *International Journal of Lexicography*, (1990), pp. 235-312.
 18. R. Baeze-Yates and B. Ribeiro-Neto, *Modern information retrieval*, (Addison-Wesley ACM Press Reading MA 1999).
 19. B. Larsen and C. Aone, Fast and effective text mining using lineartime document clustering, in *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (1999), pp. 16-22.