On the Combination of Textual and Semantic Descriptions for Automated Semantic Web Service Classification

Ioannis Katakis and Georgios Meditskos and Grigorios Tsoumakas and Nick Bassiliades and Ioannis Vlahavas

Abstract Semantic Web services have emerged as the solution to the need for automating several aspects related to service-oriented architectures, such as service discovery and composition, and they are realized by combining Semantic Web technologies and Web service standards. In the present paper, we tackle the problem of automated classification of Web services according to their application domain taking into account both the textual description and the semantic annotations of OWL-S advertisements. We present results that we obtained by applying machine learning algorithms on textual and semantic descriptions separately and we propose methods for increasing the overall classification accuracy through an extended feature vector and an ensemble of classifiers.

1 Introduction

Semantic Web services (SWSs) aim at making Web services (WSs) machine understandable and use-apparent, utilizing Semantic Web technologies (e.g. OWL-S[1], WSMO[2], SAWSDL [11]) and tools (e.g. Description Logic (DL) reasoners [2]) for service annotation and processing.

The increasing number of available WSs has raised the need for their automated and accurate classification in domain categories that can be beneficial for several tasks related to WSs, such as:

Ioannis Katakis · Georgios Meditskos · Grigorios Tsoumakas · Nick Bassiliades · Ioannis Vlahavas
Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki 54124, Greece,
e-mail: {katak, gmeditsk, greg, nbassili, vlahavas}@csd.auth.gr

[1] http://www.daml.org/services/owl-s/

[2] http://www.wsmo.org/

- *Discovery.* The effectiveness and efficiency of service discovery algorithms can be improved using WS classification by filtering out services that do not belong to the domain of interest.
- *Composition.* The classification of WSs can be used in order to increase the accuracy of WS composition by examining only the domain-relevant services in each step of the service workflow generation process.
- *Management.* The management of large number of WSs in repositories (UDDI[3]) is more effective when services are organized into categories. Furthermore, automated service classification can be utilized during the process of registering WSs in repositories by recommending service categorizations to the users.

This work presents a method for the automatic classification of SWSs based on their OWL-S Profile instances (advertisements). A Profile instance provides descriptive information about the service, such as textual description, as well as semantic annotations of WS's *inputs*, *outputs* (annotated with ontology concepts), *preconditions*, *effects* (expressed using a rule formalism, e.g. SWRL[4]), *non-functional properties*, etc. The definition of complete WS advertisements requires people with elaborate skill on description creation. Such advertisements are seldom encountered in practice. We therefore focus on the minimum piece of information that always exists in WS advertisements: the textual description and the I/O annotation concepts, also called *signatures*.

The main contribution of our work can be summarized in the following points:

1. We study the utility of four representation models for automated WS classification based on textual descriptions and signatures using a set of five different machine learning classifiers.
2. We propose and evaluate three different approaches for combining textual and semantic features. We consider such a combination vital since the textual descriptions are unable by itself to capture service's semantics, and that signatures are sometimes not sufficient to identify the service's application domain. Our experiments have shown that such a combination achieves the best overall accuracy through an ensemble of classifiers.
3. For evaluation purposes we create six different versions of our dataset of WSs that we make available online.

The rest of the paper is structured as follows. In the following section (Section 2) we briefly comment on related work on WS classification. Next (Section 3) we present four different representation methods of WSs for classification and three different approaches for combining them (Section 4). All of these methods are evaluated in Section 5 while Section 6 concludes the paper with plans for future work.

[3] http://www.oasis-open.org/committees/uddi-spec

[4] http://www.w3.org/Submission/SWRL/

2 Related Work

During the last years, a considerable effort was made for developing automatic or semi-automatic methods for classifying WSs into their application domain. In [3], WSDL[5] text descriptions are used in order to perform automatic classification of WSs using Support Vector Machines (SVMs) [21]. Many approaches [7, 13, 18, 6] use structured text elements from various WSDL components (e.g. operations) as input to various classification methods like naive Bayes [7, 13], SVMs [7], decision trees [18] or even ensemble of classifiers [6, 18]. The main disadvantage of such approaches is that no semantic information is taken into account that, as we discuss in this paper, can be considerably beneficial for classification.

In [5], the classification of WSs is based on OWL-S advertisements and it is achieved tby calculating the similarities of I/O annotation concepts between the unclassified WS and a set of preclassified WSs for each class. The predicted class is the one with the greatest overall similarity. The main disadvantage of this approach is that the representation is not flexible enough in order to be used with any machine learning algorithm and that the text of the description is ignored. We provide evaluation results that prove the utility of even short textual descriptions that may appear in the description of the WS advertisement.

A similar task to classification is SWS matchmaking. In this case a query WS description is given in order to find a set of similar WSs [9, 10].

3 Vector-based Representation of OWL-S Advertisements

This section describes a number of approaches for representing the OWL-S advertisement of a WS as a feature vector. Given a collection of labeled WSs, the corresponding feature vectors along with the labels will constitute the training examples for the machine learning algorithm.

3.1 Textual Description

Textual descriptions can be obtained from the `textDescription` property of OWL-S advertisements, from semantically enhanced UDDI registries [14], or even from the WSDL grounding of OWL-S advertisements. We represent an advertisement as a vector $\mathbf{T}_i = \left(t_{(i,1)}, \ldots, t_{(i,|V|)}\right)$ where $|V_T|$ is the size of the vocabulary V_T (the set of all distinct words in the textual descriptions of all WSs in the collection) and $t_{(i,j)}$ is the weight of the *j*-th word of the vocabulary for the *i*-th WS. A popular way to select weight for document classification is to use $t_{(i,j)} = 1$ if the *j*-th word appears in the document or $t_{(i,j)} = 0$ if not. The intuition behind this representation is

[5] www.w3.org/TR/wsdl

that the human entered textual description will contain words that will discriminate one category from another.

3.2 Ontology Imports

An OWL-S advertisement contains import declarations that denote the ontologies that are used for signature (I/O) annotations. It could be argued that these import declarations can be used in the classification procedure, following the intuition that advertisements with similar import declarations might belong to the same thematic category. To investigate this assumption, we introduce the *OntImp* vector representation of an advertisement. Let V_O be the ontology vocabulary, that is, the set of all distinct ontologies that are imported by the advertisements, taking into consideration import closures. The vector-based representation of an advertisement in the *OntImp* approach is of the form $\mathbf{O}_i = \left(o_{(i,1)}, \ldots, o_{(i,|V_O|)}\right)$, where $o_{(i,j)} = 1$, if the j-th ontology is imported (directly or indirectly) by the advertisement of the i-th WS, or $o_{(i,j)} = 0$, otherwise.

3.3 Syntactic and Semantic Signature

The signature of a WS encapsulates important domain knowledge that can be used in the classification procedure. Users annotate the I/O WS parameters with ontology concepts, defining abstractly the domain of the parameters using formal semantic descriptions. The relationships among the I/O concepts, such as *exact*, *plugin* and *subsume* [15], are determined using an ontology reasoner that computes the subsumption hierarchy of the underlying ontologies. Therefore, if two WS signatures have all or some of their I/O parameters relevant, according to some degree of relaxation, then they can possibly belong to the same category. In order to investigate the impact of the WSs' signatures, we have implemented two versions of signature-based classification; one based on the syntax (*SynSig*), treating the annotation concepts as plain text, and another based on the semantics (*SemSig*) of I/O concept annotations utilizing an OWL DL reasoner.

Syntactic Signature. Let V_C be the vocabulary of the ontology concepts, that is, the set of the distinct concepts that are used as I/O annotations in the advertisements. The representation of an advertisement in the *SynSig* approach is of the form $\mathbf{N}_i = \left(n_{(i,1)}, \ldots, n_{(i,|V_C|)}\right)$, where $n_{(i,j)} = 1$, if the j-th ontology concept is used as an input or output annotation by the advertisement of the i-th WS, or $n_{(i,j)} = 0$, otherwise.

Semantic Signature. The vector-based representation of an advertisement in the *SemSig* approach is of the form $\mathbf{S}_i = \left(s_{(i,1)}, \ldots, s_{(i,|V_C|)}\right)$. The weights are again binary, but they are selected as depicted in Algorithm 1. More specifically, if the j-th concept is referenced directly in the description of the i-th WS (line 4), or there is an annotation concept k in the i-th WS, such that j is equivalent ($j \equiv k$), superclass

($j \sqsupseteq k$) or subclass ($j \sqsubseteq k$) to k (line 8), then $s_{(i,j)} = 1$. Otherwise, if there is not such a concept k or the concepts j and k are disjoint (line 6), then $s_{(i,j)} = 0$.

Algorithm 1: semSigVector

```
Input: The ontology concept vocabulary $V_C$, the WS description $i$ and the DL reasoner $R$
Output: The weighted vector $S_i$
1  Set $inouts \leftarrow i$.inputs $\cup$ $i$.outputs;
2  $S_i \leftarrow [0,..,0]$;
3  forall $j \in inouts$ do
4      $S_i[V_C$.index(j)] $\leftarrow 1$;
5      forall $k \in V_C$ do
6          if $R(j \sqcap k \sqsubseteq \perp)$ then
7              continue;
8          if $R(j \equiv k) \vee R(j \sqsubseteq k) \vee R(k \sqsubseteq j)$ then
9              $S_i[V_C$.index(k)] $\leftarrow 1$
10 return $S_i$
```

4 Combining Text and Semantics

The WS classification based only on the semantics of signatures (*SemSig*) is not always sufficient to determine the category, since the semantic information that can be expressed is limited with respect to the details that can be captured. In that way, two WSs with different domains may have the same signature, for example, a car and an apartment rental service. In such cases, the textual descriptions can be used in order to perform a more fine-grained categorization.

On the other hand, the classification of WSs using only the text descriptions (or the *SynSig* approach) is not sufficient enough. Plain text is unable to give a formal and machine-processable semantic specification to the annotated resources that would enable the use of inference engines. Therefore, the semantic descriptions can provide an explicit and shared terminology to describe WSs, offering a more formal representation with an underlying formalization.

We argue that the combination of textual and semantic information can lead to more descriptive representations of WS advertisements. In the following sections, we propose two methods for such a combination.

4.1 Extended Feature Vector

In this case we merge the textual and syntactic / semantic vector into one, expecting from the classifier to learn relationships between textual features, syntactic/semantic features and categories. We denote the vector that represents the combination of the textual description (**T**) and the syntactic signature *TextSynSig* (**N**) as:

$$(\mathbf{TN})_i = (t_{(i,1)}, \ldots, t_{(i,|V_T|)}, n_{(i,1)}, \ldots, n_{(i,|V_C|)}) \tag{1}$$

and the one that represents the combination of the textual description and the semantic signature *TextSemSig* (**S**) as:

$$(\mathbf{TS})_i = \left(t_{(i,1)}, \ldots, t_{(i,|V_T|)}, s_{(i,1)}, \ldots, s_{(i,|V_C|)}\right) \tag{2}$$

4.2 Classifier Ensemble

Many machine learning algorithms can output not only the predicted category for a given instance but also the probability that the instance will belong to each category. Having two classifiers trained, one on textual features $H_T(d,\lambda) \rightarrow [0,1]$ and one on semantic features $H_S(d,\lambda) \rightarrow [0,1]$ that output the probability that the WS d will belong to category λ, we define two different decision schemas. If L is the set of all categories then let $h_T = \arg\max_{\lambda \in L} H_T(d,\lambda)$ and $h_S = \arg\max_{\lambda \in L} H_S(d,\lambda)$ be the decisions of H_T and H_S respectively and h_E the decision of the ensemble. The first schema (E_{max}) just selects the decision of the most confident classifier. In other words, $h_E = h_T$ if $H_T(d,h_T) \geq H_S(d,h_S)$ or $h_E = h_S$ otherwise. The second schema (E_{avg}) averages the probabilities over both classifiers for a given category and then selects the category with the maximum average:

$$h_E = \arg\max_{\lambda \in L} \left(\frac{H_T(d,\lambda) + H_S(d,\lambda)}{2} \right) \tag{3}$$

5 Evaluation

In order to evaluate the aforementioned methodologies and study their advantages and limitations we have applied them into a dataset of pre-classified WSs.

5.1 Experimental Setup

We used the OWLS-TC ver. 2.2 collection[6] that consists of 1007 OWL-S advertisements, without any additional modification. The advertisements define profile instances with simple atomic processes, without pointing to physical WSDL descriptions. Therefore, in our experiments we did not incorporate any WSDL construct. The textual description of each advertisement consists of the service name and a short service description. The WS I/O parameters are annotated with concepts from a set of $23(=|Vo|)$ ontologies that the collection provides. The advertisements are also preclassified in seven categories, namely *Travel*, *Education*, *Weapon*, *Food*, *Economy*, *Communication*, and *Medical*. Please note that this collection is an artificial one. However, it is the only publicly available collection with a relatively large number of advertisements, and it has been used in many research efforts. After a preprocessing of the collection we obtained $|V_c| = 395$ and $|V_T| = 456$. All different versions of the resulting dataset are available online in Weka format at http://mlkd.csd.auth.gr/ws.html.

In all of experiments we used the 10-fold cross validation evaluation procedure and the Pellet DL reasoner [19] in order to compute the subsumption hierarchies of the imported ontologies. In order to obtain classifier-independent results, we have tested all of the approaches discussed in the previous section with 5 different classifiers: 1) the Naive Bayes (NB) [8] classifier, 2) the Support Vector Machine (SVM) (SMO Implementation [16]), 3) the k Nearest Neighbor (kNN) classifier [1], 4) the RIPPER rule learner [4] and 5) the C4.5 decision tree classifier [17]. We used algorithms from different learning paradigms, in order to cover a variety of real-world application requirements. We used the Weka [22] implementations of all algorithms with their default settings. The kNN algorithm was executed with $k = 3$. E_{max} and E_{avg} are implemented by training two classifiers of the *same* type (one from *Text* and one from *SemSig* representation) and using the combination schemes described in Section 4. It would be interesting to study the combination of models of different type but we consider this study out of the scope of this paper.

5.2 Results and Discussion

Table 1 presents the predictive accuracy for all methods and classifiers. With bold typeface we highlight which method performs best for a specific classifier while we underline the accuracy of the best performing classifier for each method. We first notice the high performance of the SVM which achieves the best predictive accuracy for almost all cases. The second best performance is achieved by C4.5.

Considering the different representation methods we first observe that the accuracy of the *Text* representation reaches high levels (outperforming *SynSig* and *OntImp*) even with this small amount of text from the OWL-S `textDescription`

[6] http://projects.semwebcentral.org/projects/owls-tc/

property. This is probably due to the existence of characteristic words for each category. The *OntImp* vector-based representation performs the worst mainly because there are general-purpose ontologies in the collection that are imported by domain unrelated advertisements. Moreover the *SynSig* approach despite its simplicity (without the use of Pellet) achieves a decent performance. However, the better performance of *SemSig* over *SynSig* stretches the importance of the inferencing mechanism. By employing a reasoner, we are able to deduce more semantic relationships among the annotation concepts, beyond simple keyword matches, such as equivalent ($\equiv$) or subsumed ($\sqsubseteq$) concepts.

By studying the results of the enhanced representations *TextSynSig* and *TextSemSig* we observe that both approaches outperform their corresponding basic representations (*Text* and *SynSig* for the former and *Text* and *SemSig* for the latter). This fact is an indication that the classifier successfully takes advantage of both textual and syntactic / semantic features .

Another fact that stretches the importance of combining text and semantics is the accuracy of the two ensemble methods E_{max} and E_{avg} that present the best overall performance. E_{max} and E_{avg} outperform *TextSemSig* probably because they build two experts (one from text and one from semantics) while *TexSemSig* builds one model that learns to combine both set of features.

Method / Classifier	NB	SVM	*k*NN	C4.5	Ripper	AVG
Text	90.37	94.04	91.96	90.17	87.98	90.90
OntImp	60.68	79.64	77.16	80.04	74.98	74.50
SynSig	84.51	94.04	89.37	87.19	86.59	88.34
SemSig	85.80	**96.92**	90.37	93.55	90.86	91.50
TextSynSig	89.97	95.73	92.85	90.57	87.69	91.36
TextSemSig	**91.96**	96.52	93.74	93.15	91.96	93.47
E_{max}	91.76	95.43	94.34	95.63	**92.95**	94.02
E_{avg}	**91.96**	96.23	**94.64**	**95.93**	92.85	94.12
AVG	85.89	93.44	90.55	90.78	88.23	

Table 1 Predictive accuracy of all methods and classifiers

6 Conclusions and Future Work

In this paper we presented an approach for automated WS classification based on OWL-S advertisements. We have presented several ways of representing semantic descriptions as vectors, each one with different semantic capabilities. In general, the exploitation of the semantic signature can lead to better classification accuracy than using the syntactic signature of a WS. Furthermore, we elaborated on two approaches for combining the text- and semantic-oriented vectors in order to exploit the descriptive capabilities of each paradigm, increasing the classification accuracy. Note, that our methodology can be extended to other SWS standards, such as SAWSDL.

Our classification approach can be extended in two directions. Firstly, the *SemSig* representation can be extended in order to incorporate also non-binary vectors, using as weights the similarities that are computed by concept similarity measures [12]. In that way, we will be able to define different degrees of relaxation in the representation. Secondly, it would be interesting to experiment with multilabel classification methods [20] for collections of SWSs that belong to more than one category.

Acknowledgements This work was partially supported by a PENED program (EPAN M.8.3.1, No. 03EΔ73), jointly funded by EU and the General Secretariat of Research and Technology.

References

1. Aha, D., Kibler, D.: Instance-based learning algorithms. Machine Learning **6**, 37–66 (1991)
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F.: The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
3. Bruno, M., Canfora, G., Penta, M.D., Scognamiglio, R.: An approach to support web service classification and annotation. In: Proceedings IEEE International Conference on e-Technology, e-Commerce and e-Service, pp. 138–143. Washington, DC (2005). DOI http://dx.doi.org/10.1109/EEE.2005.31
4. Cohen, W.W.: Fast effective rule induction. In: Proceedings 12th International Conference on Machine Learning, pp. 115–123 (1995)
5. Corella, M., Castells, P.: Semi-automatic semantic-based web service classification. In: J. Eder, S. Dustdar (eds.) Business Process Mangement Workshops, Springer Verlag Lecture Notes in Computer Science, vol. 4103, pp. 459–470. Vienna, Austria (2006)
6. Heß, A., Johnston, E., Kushmerick, N.: ASSAM: A tool for semi-automatically annotating semantic web services. In: Proceedings 3rd International Semantic Web Conference (2004)
7. Hess, A., Kushmerick, N.: Learning to attach semantic metadata to web services. In: Proceedings International Semantic Web Conference (ISWC'03), pp. 258–273 (2003)
8. John, G.H., Langley, P.: Estimating continuous distributions in bayesian classifiers. In: Proceedings 11th Conference on Uncertainty in Artificial Intelligence, pp. 338–345. Morgan Kaufmann, San Mateo (1995)
9. Kiefer, C., Bernstein, A.: The creation and evaluation of isparql strategies for matchmaking. In: M. Hauswirth, M. Koubarakis, S. Bechhofer (eds.) Proceedings 5th European Semantic Web Conference, LNCS. Springer Verlag, Berlin, Heidelberg (2008). URL http://data.semanticweb.org/conference/eswc/2008/papers/133
10. Klusch, M., Kapahnke, P., Fries, B.: Hybrid semantic web service retrieval: A case study with OWLS-MX. In: International Conference on Semantic Computing, pp. 323–330. IEEE Computer Society, Los Alamitos, CA (2008). DOI http://doi.ieeecomputersociety.org/10.1109/ICSC.2008.20
11. Kopecký, J., Vitvar, T., Bournez, C., Farrell, J.: Sawsdl: Semantic annotations for wsdl and xml schema. IEEE Internet Computing **11**(6), 60–67 (2007). DOI http://dx.doi.org/10.1109/MIC.2007.134
12. Meditskos, G., Bassiliades, N.: Object-oriented similarity measures for semantic web service matchmaking. In: Proceedings 5th IEEE European Conference on Web Services (ECOWS'07), pp. 57–66. Halle (Saale), Germany (2007)

13. Oldham, N., Thomas, C., Sheth, A., Verma, K.: Meteor-s web service annotation framework with machine learning classification. In: Proceedings 1st International Workshop on Semantic Web Services and Web Process Composition (SWSWPC'04), pp. 137–146 (2005)
14. Paolucci, M., Kawamura, T., Payne, T.R., Sycara, K.P.: Importing the semantic web in uddi. In: Revised Papers from the International Workshop on Web Services, E-Business, and the Semantic Web (CAiSE'02/WES'02), pp. 225–236. Springer-Verlag, London, UK (2002)
15. Paolucci, M., Kawamura, T., Payne, T.R., Sycara, K.P.: Semantic matching of web services capabilities. In: Proceedings 1st International Semantic Web Conference on The Semantic Web (ISWC'02), pp. 333–347. Springer-Verlag, London, UK (2002)
16. Platt, J.: Machines using sequential minimal optimization. In: B. Schoelkopf, C. Burges, A. Smola (eds.) Advances in Kernel Methods - Support Vector Learning. MIT Press (1998). URL http://research.microsoft.com/j̃platt/smo.html
17. Quinlan, R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA (1993)
18. Saha, S., Murthy, C.A., Pal, S.K.: Classification of web services using tensor space model and rough ensemble classifier. In: Proceedings 17th International Symposiumon Foundations of Intelligent Systems (ISMIS'08), pp. 508–513. Toronto, Canada (2008)
19. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical owl-dl reasoner. Web Semant. **5**(2), 51–53 (2007). DOI http://dx.doi.org/10.1016/j.websem.2007.03.004
20. Tsoumakas, G., Katakis, I.: Multi-label classification: An overview. International Journal of Data Warehousing and Mining **3**(3), 1–13 (2007)
21. Vapnik, V.N.: The nature of statistical learning theory. Springer-Verlag, NY, USA (1995)
22. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd Edition. Morgan Kaufmann Publishers Inc., San Francisco, CA (2005)