CSR: Discovering Subsumption Relations for the Alignment of Ontologies

Vassilis Spiliopoulos[1, 2], Alexandros G. Valarakos[1], and George A. Vouros[1]

[1]AI Lab, Information and Communication Systems Engineering Department, University of the Aegean, Samos, 83 200, Greece
{vspiliop, alexv, georgev}@aegean.gr

[2] Institution of Informatics and Telecommunications, NCSR "Demokritos", Greece

Abstract. For the effective alignment of ontologies, the computation of equivalence relations between elements of ontologies is not enough: Subsumption relations play a crucial role as well. In this paper we propose the "Classification-Based Learning of Subsumption Relations for the Alignment of Ontologies" (*CSR*) method. Given a pair of concepts from two ontologies, the objective of *CSR* is to identify patterns of concepts' features that provide evidence for the subsumption relation among them. This is achieved by means of a classification task, using state of the art supervised machine learning methods. The paper describes thoroughly the method, provides experimental results over an extended version of benchmarking series and discusses the potential of the method.

Keywords: ontology alignment, subsumption, supervised machine learning

1 Introduction

In spite of the fact that ontologies provide a formal and unambiguous representation of domain conceptualizations, it is rather expectable to deal with different ontologies describing the same domain of knowledge, introducing heterogeneity to the conceptualization of the domain and difficulties in integrating information.

Although many efforts [1] aim to the automatic discovery of equivalence relations between the elements of ontologies, in this paper we conjecture that this is not enough: To deal effectively with the ontologies' alignment problem, we have to deal with the discovery of subsumption relations among ontology elements. This is particularly true, when we deal with ontologies whose conceptualizations are at different "granularity levels": In these cases, the elements (concepts and/or properties) of an ontology are more generic than the corresponding elements of another ontology. Although subsumption relations between the elements of two ontologies may be deduced by exploiting equivalence relations between other elements (e.g., a concept C_1 is subsumed by all subsumers of C_2, if C_1 is equivalent with a concept C_2), in the extreme cases where no equivalence relations exist, this can not be done. In any case, we conjecture that the discovery of subsumption relations between elements of different ontologies can enhance the discovery/filtering of equivalence relations, and vise-versa, augmenting the effectiveness of our ontology alignment and merging methods.

This is of great importance when dealing with real-world ontologies, where, as it is also stated in the conclusions of the Consensus Track of OAEI 06 [2], current state of the art systems "confuse" subsumption relations with equivalence ones.

To make the above claims more concrete, let us consider the ontologies depicted in Fig. 1. These specify the concept `Citation` in O_1 (which is equivalent to the concept `Reference` in O_2), and `Publication` in O_2 (which is equivalent to the concept `Work` in O_1). Each of these ontologies elaborates on the specification of distinct concepts: O_2 elaborates on the concept `Publication` and O_1 on the concept `Citation`. Furthermore, as shown in Fig.1, concepts are related among themselves via object properties whose lexicalizations differ: For instance, in O_2, the concept `Reference` is related via the object property `of` with the concept `Publication`, while in O_1, the corresponding concept `Citation` is related via the object property `to` with the concept `Work`. Given these ontologies, and given that equivalent properties in the two ontologies do not have the same lexicalization, and that non-equivalent concepts do have the same lexicalization, we may distinguish two cases:

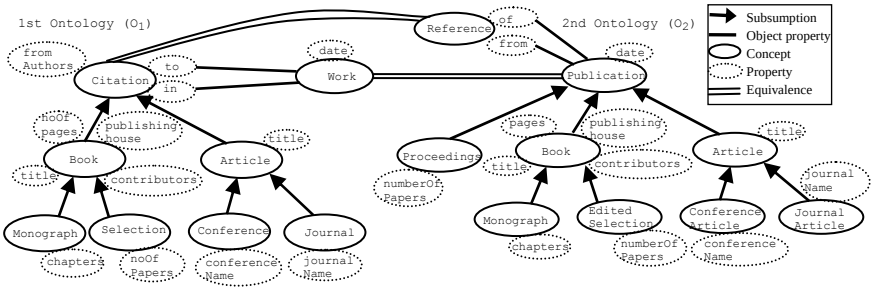

Fig. 1. Example ontologies for assessing the subsumption relation between concepts.

In case that the equivalencies between the concepts of the two ontologies are not known, conclusions concerning subsumption relations between the concepts of the two ontologies cannot be drawn by a reasoning mechanism: This case clearly shows the need to discover both equivalence and subsumption relations between the concepts of the source ontologies.

In case the equivalences between the concepts of these two ontologies are known (the automatic discovery of these equivalencies is not a trivial task), one may deduce subsumption relations between the subsumees of these concepts. Specifically, a reasoning engine shall deduce that concepts that share the same lexicalization and the same properties are equivalent, which is wrong in our case: For instance, given that the properties of the concept `Book` of O_1 are pair wise equivalent to the properties of the concept `Book` of O_2, a reasoning service may wrongly assess that the concept `Book` of O_1 is equivalent to the concept `Book` of O_2: However `Book` from O_1 specifies book citations, while `Book` from O_2 specifies book publications. Therefore, it seems that although the discovery of subsumption relations among the elements of distinct

ontologies must be done with respect to the known equivalence relations, a reasoning mechanism does not suffice for this purpose.

What is clearly needed is a method that shall discover subsumption relations between concept pairs of two distinct ontologies, separately from subsumptions and equivalencies that can be deduced by a reasoning mechanism. For instance, the concept `Book` of O_1 (respectively, O_2) must be assessed to be subsumed by the concept `Reference` (respectively, `Work`) of O_2 (respectively, O_1), without assessing that it is equivalent to the concept `Book` of this ontology, even if their properties and labels are identical. By admitting the later wrong mapping, numerous wrong subsumption relations can be deduced, e.g., that `Book` in O_1 is subsumed by the `Publication` in O_2 (`Book` specifies book citations, while `Publication` the publications themselves).

This paper deals with discovering subsumption relations between concepts of two distinct ontologies. This is done by using the "Classification-Based Learning of Subsumption Relations for the Alignment of Ontologies" (CSR) method. CSR computes subsumption relations between concept pairs of two distinct ontologies by means of a classification task, using state of the art supervised machine learning methods. The classification mechanisms proposed exploit two types of concepts' features: Concept properties, and terms extracted from labels, comments, properties and instances of concepts. Specifically, given a pair of concepts from the two source ontologies, the classification method "locates" a hypothesis concerning concepts' relation, which best fits to the training examples [3], generalizing beyond them. Concept pairs are represented as feature vectors of length equal to the number of the distinct features of source and target ontologies: In case features correspond to concept properties, properties that are equivalent (i.e., properties with equivalent meaning) correspond to the same vector component. In case features are terms, then terms with the same surface appearance correspond to the same vector component. It must be pointed that the examples for the training of the classifiers are being generated by exploiting the known subsumption and equivalence relations in both source ontologies, considering each source ontology in isolation.

The machine learning approach has been chosen since (a) there are no evident generic rules *directly* capturing the existence of a subsumption relation between ontology elements, and (b) concept pairs of the same ontology provide examples for the subsumption relation, making the method self-adapting to idiosyncrasies of specific domains, and non-dependant to external resources. The conjecture is that, if the supervised learning method generalizes successfully from the training examples, then the learned model shall capture the "patterns", in terms of the chosen features (i.e., properties or terms), for the discovery of subsumption relations that can not be deduced by a reasoning mechanism.

The rest of the paper is structured as follows: Section 2 states the problem and presents works that are most closely related to our approach. Section 3 provides background knowledge concerning the learning and classification methods used. Section 4 presents the proposed classification-based method for subsumption discovery. Section 5 presents and thoroughly discusses the experimental settings, as well as the results. Section 6 concludes the paper by pointing out the key points of our method and sketching further work for the improvement of the method.

2 Problem Statement and Related Work

2.1 Problem Statement

An ontology is a pair $O=(S, A)$, where S is the ontological signature describing the vocabulary (i.e., the terms that lexicalize ontology elements) and A is a set of ontological axioms, restricting the intended meaning of the terms included in the signature. In other words, A includes the formal definitions of ontology elements that are lexicalized by natural language terms in S. Subsumption relations are ontological axioms included in A. Distinguishing between concepts and properties, we consider a partition of S comprising the sets S_p and S_c, denoting the sets of terms lexicalizing ontology properties and ontology concepts, respectively. Let also T be the set of distinct terms that are in S, or that are extracted from labels, comments or instances of ontology elements.

Ontology mapping from a source ontology $O_1=(S_1,A_1)$ to a target ontology $O_2=(S_2,A_2)$ is a morphism $f:S_1 \rightarrow S_2$ of ontological signatures specifying elements' equivalences, such that $A_2 \models f(A_1)$, i.e., all interpretations that satisfy O_2's axioms also satisfy O_1's translated axioms. However, considering different types of relations between ontology elements, the ontology mapping problem can be stated as follows: Classify any pair (C^1,C^2) of elements of the input ontologies, such that C^i is a term in S_i, $i=1,2$, to any of the following relations, consistently: equivalence ($\equiv$), subsumption (inclusion) ($\sqsubseteq$), mismatch ($\perp$) and overlapping ($\sqcap$). By doing this, ontologies O_1 and O_2 can be aligned, resulting to a new consistent and coherent ontology.

In this paper we deal with the *subsumption computation problem* which, given the above generic problem, is as follows: Given (a) a source ontology $O_1=(S_1,A_1)$ and a target ontology $O_2=(S_2,A_2)$ such that $S_1=S_{1c} \cup S_{1p}$ and $S_2=S_{2c} \cup S_{2p}$, (b) the set $T_1 \cup T_2$ of distinct terms that appear in both ontologies (considering terms with the same surface appearance to be "equivalent" in meaning), and optionally (c) a morphism $f:S_{1p} \rightarrow S_{2p}$ from the lexicalizations of the properties of the source ontology to the lexicalizations of the properties of the target ontology (specifying properties' equivalences), classify each pair (C^1,C^2) of concepts, where C^1 is a term in S_{1c} and C^2 is a term in S_{2c}, to two distinct classes: To the "subsumption" ($\sqsubseteq$) class, or to the class "*R*". The latter class denotes pairs of concepts that are not known to be related via the subsumption[1] relation, or that are known to be related via the equivalence, mismatch or overlapping ones.

2.2 Related Work

Due to the evolving nature of ontologies, to the large number of elements that they comprise, and to the importance of the ontology alignment task, there are many research efforts towards automating this task. The majority of these methods focus on discovering equivalence relations between ontology elements [1] (e.g., concepts and properties). As a result, there has been a dramatic increase in the efficacy and effi-

[1] This means that a pair of concepts belonging to "*R*" may belong to the subsumption relation.

ciency of the methods that locate equivalences among ontology elements, while subsumption relations have not been thoroughly studied.

Concerning the computation of subsumption relations, related works have strong dependence on external resources, such as WordNet, domain ontologies or text corpora. A limitation that does not apply in the method proposed in this paper.

The method proposed in [4] transforms the mapping problem into a satisfiability problem, by taking into account the hierarchical relations between WordNet senses, along with the lexical and structural knowledge of the input ontologies.

Another related approach [5] introduces the WordNet Description Logics (WDL) language so as to align two different ontologies. WordNet is treated as an intermediate ontology. Similarly, in [6], [7] the authors propose the exploitation of background knowledge in the form of domain ontologies.

The authors in [8] loosen the formal constraints of the subsumption relation by exploiting hits returned by Google. Two more Google-based approaches [9], [10] exploit the so called Hearst patterns and test their validity by exploiting the returned hits.

Most machine learning based approaches aim to the discovery of equivalence relations between ontology elements and do not deal with subsumption relations as we do in this work. To the best of our knowledge, the most relevant machine learning technique is presented in [11]. Specifically, the authors propose a method based on Implication Intensity theory, which is a probabilistic model of deviation from statistical independence. The method takes as input a hierarchy of concepts and a set of documents, each one indexed under a specific concept. Then, the proposed model is applied in order to locate strong derivations between sets of terms that appear in the documents and as a consequence between their indexed concepts.

In this paper we consider the subsumption computation problem as a binary classification problem, where a classifier has to assess whether a pair of concepts belongs to the subsumption relation. As it will be explained, the semantics of the input ontologies are exploited in order the method to generate the appropriate examples for the training of the classifier. This makes the proposed method dependent only from the source ontologies and independent from any third/external domain resource.

3 Classification and Inductive Learning

Classification is one of the main problems addressed within the machine learning discipline. It concerns the classification of example cases into one of a discrete set of classes. When the number of classes is restricted to two, the problem is referred to as a binary classification problem. More accurately, the binary classification problem is defined as follows: Given a set of m examples (x^j, y^j), j=1, 2, … , m (the training dataset) of vectors x^j sampled from some distribution D, the *output* is a function $c:R^n \rightarrow \{0,1\}$ (classifier) which classifies additional samples x^k sampled from the same distribution D to the classification classes $\{0,1\}$. It holds that $x^j \in R^n$ and $y^j \in \{0,1\}$. The i-th component of vector x^j is termed the *feature i*, X_i^j of the x^j sample.

In the context of studying the subsumption computation problem, we have used specific implementations of well studied classifiers: (a) *Probabilistic classifiers* spec-

ify the function *c* as a probabilistic function, assessing the probability $p(x^j, y^j)$ that the document x^j falls within a category y^j. From this category we have selected a Naïve Bayes (Nb) classifier. (b) *Memory-based classifiers* store the training data in memory and when a new instance is encountered, similar instances are retrieved from their memory and used for the instance classification. We used the k-nearest neighbor (Knn) with value of k=2. (c) *Support Vector Machines (SVMs)* based classifiers map input vectors to a higher dimensional space where a maximal separating hyperplane is constructed. The transformation of the data to the new space is made through functions called kernels. We have selected the libSVM [12] implementation with its default values and radial basis function as kernel. (d) *Decision Tree classifiers* exploit a tree structure in which each interior node corresponds to a feature. The branch from a node to a child represents a possible value of that feature, and a leaf node represents the classification class given the values of the features represented by the path from the root. Weka's j48 [3] is the implementation of the widely used state of the art C4.5 decision tree learning algorithm that we have used in this work.

4 The *CSR* Method

4.1 Description of the overall *CSR* method

The discrete steps of the CSR method, as depicted in Fig. 2, are the following:

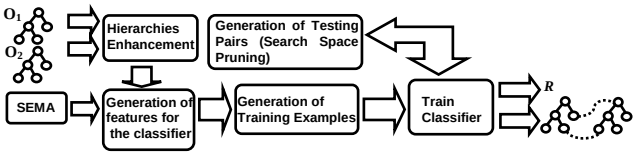

Fig. 2. Overview of the *CSR* method.

- Reasoning services are being used for inferring all facts according to ontologies' specification semantics: The objective of this step is to compute implied subsumption and equivalence relations between existing ontology elements. This is a necessary step as it affects the generation of the training dataset (subsection 4.3).
- Currently CSR exploits two types of features: Concepts' properties and terms appearing in the "vicinity" of concepts. In both cases features are generated by gathering all discrete properties or terms from both ontologies. This is further detailed in the next subsection.
- The sets of training examples are being generated according to the rules defined in subsection 4.3. The balancing of the training dataset is an important issue that is being tackled in this step, as well.
- The classifier is being trained using the training dataset, and

– Concept pairs are being classified by the trained classifier, pruning the search space according to the method explained in subsection 4.4.

Studying the importance of concepts' properties to assessing the subsumption relation between concepts (a) appeals to our intuition concerning the importance of properties as distinguishing characteristics of concepts, (b) it provides the basis for a method considering only properties' equivalences. This basic method can be further enhanced with the computation of equivalences between other concepts' distinguishing features (e.g., concepts in a given vicinity), and can be further combined with other alignment methods. As far as the use of terms is concerned, (a) their use for describing the intended meaning of concepts appeals to our intuition, and (b) it does not necessitate the use of any method for the discovery of equivalence relations among ontology elements. This paper studies the potential of CSR with these two types of features, while leaving further enhancements and combinations for future work.

4.2 Features Generation

Each pair of concepts (C^1,C^2) is represented by a feature vector whose components' values are as follows:

– "0", if the corresponding feature does not appear neither in C^1 nor in C^2.
– "1", if the corresponding feature appears only in C^1.
– "2", if the corresponding feature appears only in C^2.
– "3", if the corresponding feature appears in both C^1 and C^2.

As it can be noticed, feature vectors are not identical for symmetrical pairs of concepts. This allows the computation of the direction of the subsumption relation. In the case where properties are being used as features of concept pairs, given the equivalences among ontology properties computed by a morphism *f*, equivalent properties correspond to the same component of concepts' feature vectors. To compute properties' equivalences, we have used the SEMA [13] mapping tool which has been evaluated in the OAEI 2007 contest [14]. Towards discovering mappings between properties SEMA exploits: (a) Lexical information concerning names, labels and comments of ontologies' properties. (b) Properties' domain, range and hierarchy for propagating similarities among properties. Therefore, we emphasize that by "property appearance" we do not mean the occurrence of the property's lexicalization, but the occurrence of property's meaning assessed by SEMA.

In the case where terms are being used for the representation of concepts' pairs, terms are being extracted from both ontologies. Each distinct term corresponds to a specific component of the feature vector and the length of the vector is equal to the total number of all distinct terms from both input ontologies. Specifically, for each concept of the input ontologies, terms are being extracted from its "vicinity", as it is specified by the following rule: Given a concept, the method extracts terms occurring in the local name, label and comments of this concept, from all of its properties (exploiting the properties' local names, labels and comments), as well as from all of its

related concepts. Finally, terms from all instances of the corresponding concept are being extracted.

By exploiting the equivalence and disjoint relations between ontology elements, the conjunction and disjunction constructors, the appearance of a term in an element is determined by the following rules: (i) Given an ontology element, the method considers terms appearing in all its equivalent elements. (ii) If the corresponding element is defined as the disjunction (conjunction) of other elements, then the method unions (respectively, intersects) the sets of terms that appear in the constituent elements. (iii) If two elements are defined to be disjoint, then the method considers these terms that are not common in both elements.

During this step, tokenization, stemming and elimination of stop words is performed on the set of extracted words.

For example, according to the above, given the ontologies in Fig. 1 and the properties' equivalences `of`≡`to`, `from`≡`in`, `pages`≡`noOfPages`, `numberOfPapers`≡`noOfPapers` provided by SEMA, the concept pair (`Citation`,`Publication`) is represented by the feature vector (0, 3, 3, 0, 0, 0, 0, 0, 0, 0, 0, 1). The features are `date`, `of`≡`to`, `from`≡`in`, `pages`≡`noOfPages`, `publishingHouse`, `numberOfPapers`≡`noOfPapers`, `title`, `contributors`, `chapters`, `conferenceName`, `journalName` and `fromAuthors` according to the order of their appearance. Concerning case where features are terms, the feature vector is (.., 1, 3, 1, 1, 1, 1, 3, 1, 1, 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 2,..). The features in order of appearance are `citation`, `from`, `authors`, `article`, `to`, `in`, `date`, `work`, `no`, `of`, `pages`, `publishing`, `house`, `book`, `title`, `contributors`, `publication`, `proceedings`, `number`, and `reference`. All the vector components that are not shown correspond to terms that do not appear in the vicinity of any of the two concepts, so their value is set to 0.

4.3 Creating the Training Dataset

As it has been stated, training examples for classes "$\sqsubseteq$" and *R* are being generated by exploiting the source and target ontologies, according to the semantics of specifications. The basic rules for the generation of the training examples for the class "$\sqsubseteq$" are as follows:

Subsumption Relation. Include all concept pairs from both input ontologies that belong in the subsumption relation. The subsumption relation may or may not be direct. If more than one hierarchy is specified, then all hierarchies need to be exploited.

Equivalent concepts. Enrich the set of concept pairs generated by the above rule, by taking into account stated and inferred equivalence relations between concepts. In detail, for each concept pair (C^1,C^2) that belongs in the subsumption relation, and for each stated equivalence relation $C^i \equiv C^i_k$, $i \in \{1,2\}$, $k=1,2,\ldots$, then the pair (C^1,C^2_k) (or the pair (C^1_k,C^2)) belongs to the subsumption relation, as well.

Union of concepts. Enrich the set of pairs by exploiting the union construct in the definition of concepts: When one concept (e.g., the concept $C_4 \sqcup C_5$) is constructed as the union of others, and it is defined to be subsumed by another concept (e.g., by the concept C_2), then each concept in the union is subsumed by the more general one (i.e.,

it holds that $C_4 \sqsubseteq C_2$ and $C_5 \sqsubseteq C_2$). By taking into account also the equivalence rule (e.g., $C_4 \equiv C_3$), the concept C_4 can be substituted by its equivalent concept, and therefore, the pair (C_3,C_2) is included as well.

According to the open world semantics, we need to exploit the stated axioms for the generation of training examples: Therefore, in case there is not an axiom that specifies the subsumption relation between a pair of concepts (or in case this relation can not be inferred by exploiting the semantics of specifications), then this pair does not belong to the subsumption class and it is included in the generic class "*R*". The following cases summarize the rules for the generation of examples for the class "*R*":

Concepts belonging to different hierarchies. If two concepts belong to different hierarchies of the same ontology, then no explicit subsumption relation is defined among them. As a result, all pairs following this rule are characterized as training examples of the class "*R*". This set of pairs can be enriched by taking into account the stated equivalence and union relations between concepts, as explained in the case of class "$\sqsubseteq$".

Siblings at the same hierarchy level. This includes pairs of concepts that are siblings (share the same subsumer) and that are not related via the subsumption relation. As a result, all possible pairs following this rule are characterized as training examples of the class "*R*". Similarly to the first category, this category can also be enriched by exploiting concepts' equivalences and unions.

Siblings at different hierarchy levels. If any concept that is in a pair belonging in the "siblings of the same hierarchy level" category is substituted by any of its subsumees, then new pair examples are recursively generated, until the leaf concepts of the ontology are reached. These examples constitute a new category called "siblings at different hierarchy levels". Similarly with the previous categories, this one also can be enriched by exploiting the union construct of concepts and the equivalence relation between concepts.

Concepts related thought a non-subsumption relation. This includes concepts that are related via an object property and are not related with a subsumption relation. As with the previous categories, this category may also be enriched by considering unions and equivalences between concepts.

Inverse pairs of class "$\sqsubseteq$". All concepts pairs (C_2,C_1) such that C_1 subsumes C_2, but it cannot be inferred that C_2 subsumes C_1, constitute examples for the class "*R*".

As it is evidenced by the above, the number of training examples for the class "$\sqsubseteq$" are much less than the ones for class "*R*". It is very important for the performance of the classifier that the training examples for both classes to be balanced in numbers.

Being balanced in numbers, we intend that the two classes are equally represented in the training dataset. This is referred as the *dataset imbalance* problem. In the context of the classification task, various techniques have been proposed towards its solution [15]. In this work, to tackle this problem, we have adopted two alternatives: The under-sampling and the over-sampling methods.

According to the under-sampling method, all different categories of class "*R*" are equally sampled (randomly), until the selected examples are equal in numbers with the ones of class "$\sqsubseteq$". In the case of over-sampling, the method selects examples for the class "$\sqsubseteq$" randomly, until the two classes have the same number of examples.

4.4 Pruning the Search Space

Taking into account the semantics of the subsumption relation, instead of generating all possible concept pairs from both ontologies, we prune the search space by excluding pairs of concepts for which a subsumption relation can not be assessed to hold, due to the existent and currently computed relations. First we provide two short definitions: A *root concept* is every concept of the ontology that does not have a subsumer. *Root concepts* may not have sub-concepts, hence are called *unit concepts*. We consider that an ontology may include more than one subsumption hierarchies for concepts.

In order to prune the search space, the proposed algorithm firstly checks all the concepts from the first ontology and unit/root concepts of the second ontology. If a pair is not classified in the class "$\sqsubseteq$", then the hierarchy rooted by the corresponding concept of the second ontology is not being examined by the classifier. If a pair is assessed to belong to the class "$\sqsubseteq$", then the concept of the first ontology is recursively being tested with the direct subsumees of the corresponding concept in the second ontology, until either a pair is assessed to belong in the class "*R*", or until the leaf concepts are reached.

5 Experimental Results and Discussion

5.1 The Dataset

The testing dataset has been derived from the benchmarking series of the OAEI 2006 contest [14]. As our method exploits the properties of concepts (in cases where properties are used as concept pairs' features), we do not include those OAEI 2006 ontologies where concepts have no properties. The compiled corpus is available at the URL http://www.icsd.aegean.gr/incosys/csr. For each pair of ontologies we have created the gold standard ontology, including subsumption relations among concepts.

All benchmarks (101-304) except those in categories *R1-R4* (real-world cases), define the second ontology of each pair as an alteration of the same first. The benchmarks can be categorized based on their common features as follows: (a) in categories *A1-A5* (101-210, 237, 238 and 249), elements' lexicalizations of the target ontologies are altered in various ways (e.g., uppercasing, underscore, foreign language, synonyms or random strings), (b) in categories *A6-A7* (225 and 230) restrictions are removed and/or properties are modeled in more detail and/or the hierarchy is flattened, (c) in categories *F1-F2* (222, 237, 251 and 258) the hierarchies are flattened and in *F2* also random lexicalizations of all elements are introduced, and (d) categories *E1-E2* (223, 238, 252 and 259) result from *F1-F2* with expanded hierarchies.

5.2 Experiments and Results

Results show the precision and recall of the proposed method as it is applied in the different types of ontology pairs specified in subsection 5.1. Precision is the ratio

#correct_pairs_computed/*#pairs_computed* and recall is the ratio *#correct_pairs_computed*/*#pairs_in_gold_standard*.

We have run experiments for the benchmark series specified using each of the classifiers: C4.5, Knn, NaiveBayes (Nb) and Svm. For each of the classifies we have run four experiments using terms or properties as features of concept pairs, in combination with the dataset balancing method: over and under-sampling. Subsequently, we denote each type of experiment with X+Y+Z, where X is the classifier, Y is the type of features used ("Props" for properties or "Terms" for terms) and Z is the type of dataset balancing method used ("over" and "under" for over- and under-sampling). For instance, the experiment type "C4.5+Props+Over" indicates the use of the C4.5 classifier in *CSR*, exploiting properties as features, with over-sampling for balancing the training dataset.

Furthermore, the results of our method are compared to the results of a baseline classifier, which is based on the Boolean Existential Model. This classifier does not perform any kind of generalization: In order to classify a testing concept pair, it consults the vectors of the training examples of the class "⊑", and selects the first exact match. The comparison with this classifier has been performed for showing how *CSR* classifiers generalize over the training examples, learning subsumption cases not present in the training examples. Here we have to point out that both *CSR* and the baseline classifier exploit the same information. As terms or properties are being used as features, two different types of experiments have been conducted using the baselines classifier: The one with properties (Baseline+Props) and the other with terms (Baseline+Terms).

To investigate whether, given a set of equivalence relations, a reasoning mechanism suffices for the purpose of computing subsumption relations among the elements of distinct ontologies we also compare *CSR* with a Description Logics' reasoning engine[2]. In order for the reasoner to be able to infer subsumption relations between concepts of the source ontologies we specify axioms concerning only properties' equivalencies (Reasoner+Props), or alternatively, both properties' and concepts' equivalencies (Reasoner+Props+Con). At this point we must recall that when *CSR* exploits terms, then no equivalence mappings are required.

Fig. 3 and Fig. 4 depict the average precision and recall values in all types of experiments. The first important observation by looking at Fig. 3 and Fig. 4 is (as it was expected) that a reasoner cannot infer all the subsumption relations between concepts of the input ontologies. Especially, for the Reasoner+Props+Con type of experiments, the reasoner infers many false positives (precision: 41%). On the other hand, when only property mappings are exploited (Reasoner+Props) the reasoner achieves a low recall value (58%). Even in the case where the precision and recall of the equivalence mappings produced by SEMA are 100% (A1 category, Fig. 7 and Fig. 8) and the two ontologies are almost the same (only some minor axioms are suppressed in the second ontology), the reasoner achieves precision 82% and 71% depending of the type of equivalencies considered (Fig. 5). The same applies in Fig. 6 for the A1 ontologies. These results provide firm evidence that a reasoning mechanism does not suffice for the purpose of computing subsumption relations among the elements of distinct ontologies, even if equivalence relations are computed with high precision.

[2] We have used Pellet in our experiments (http://pellet.owldl.com).

This conjecture is further evidenced by the results achieved in the real world cases (R1 to R4) shown in Fig. 5 and Fig. 6. Indeed, the reasoner, in both types of experiments in R3 and R4 achieves low precision in a moderate recall. Especially, the case R4 is quite interesting, as the precision and recall of SEMA is quite high (Fig. 7 and Fig. 8). The same applies in cases F1 and A7: In F1 the precision and recall of SEMA is almost 100% (Fig. 7 and Fig. 8). In this case the reasoner exploiting both properties and concepts equivalences (Reasoner+Props+Con) achieves precision 13%, and recall 85%; the reasoner exploiting properties equivalences (Reasoner+Props) achieves precision 62% and recall 85%.

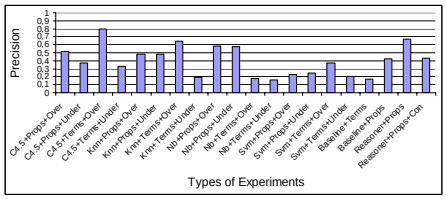

Fig. 3. Overall precision per experiment.

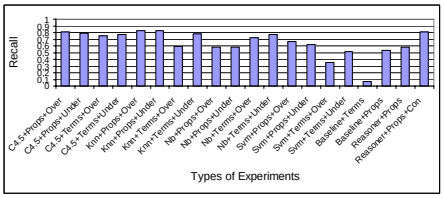

Fig. 4. Overall recall per experiment.

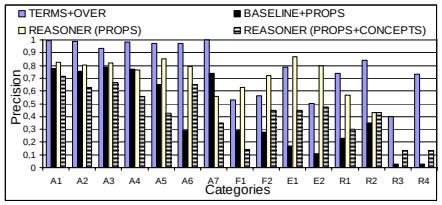

Fig. 5. Precision in all test categories.

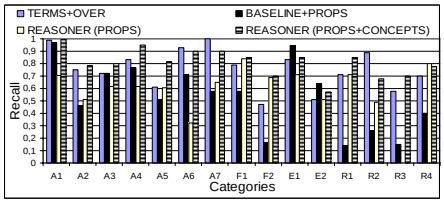

Fig. 6. Recall in all test categories.

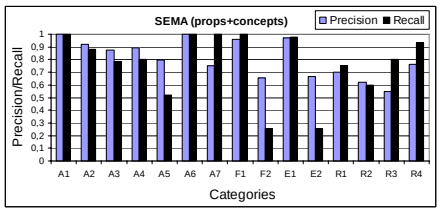

Fig. 7. SEMA's overall performance.

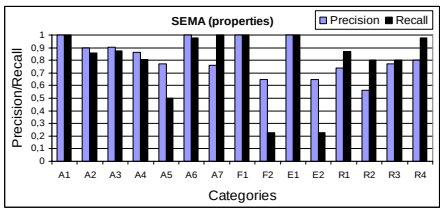

Fig. 8. SEMA for properties.

Furthermore, as Fig. 4 shows, nearly all types of experiments (with the exception of the SVM) contacted with the CSR method achieve a better recall than the baseline classifiers (Baseline+Props and Baseline+Terms). This means that classifiers do generalize, as they manage to locate subsumptions that are not in their training dataset. Moreover, for each of the classifiers there is a type of experiment in which CSR per-

forms better than the baseline classifier in terms of precision (with the exception of the SVM). A fact which is very important, since it shows that in these cases over-generalization does not take place.

As Fig. 3 shows, the C4.5 classifier exploiting terms with over-sampling (C4.5+Terms+Over) outperforms all classifiers in terms of precision. Also, as shown in Fig. 4, the same classifier achieves one of the highest recalls: Therefore, subsequently we shall focus on these type of experiments with C4.5.

The fact that C4.5+Terms+Over has the best overall performance in terms of precision and recall among all classifiers can be explained by the specific features of decision tree classifiers: (i) Disjunctive descriptions of cases, an inherent feature of decision trees, fits naturally to the subsumption computation problem. This is true since more than one features may indicate whether a specific concept pair belongs in the class "$\sqsubseteq$". (ii) Decision trees are very tolerant to errors in the training set [3]. This is true as far as the training examples, as well as the values of vector components for the representation of examples are concerned. In our case, the values of vector components may not be correct as the task for the discovery of equivalencies among properties is erroneous.

If we compare the results achieved by the best CSR classifier that exploits terms as features (C4.5+Terms+Over) to the results achieved by the reasoner that exploits the least possible input (i.e., reasoner with properties' equivalencies), as shown in Fig.3 and Fig.4, CSR performs better in terms of recall and precision. Indeed, in this case C4.5 achieves the best balance between precision (80%) and recall (78%), than any other method in the experiments, although it does not exploit equivalence mappings.

By observing Fig. 5 and Fig. 6 we see that in cases where the source ontologies differ substantially (e.g., cases A7, R1-R4) CSR with C4.5 exploiting terms and over-sampling (C4.5+Terms+Over) not only has the higher precision, but also is among the highest in terms of recall. In any case CSR achieves a good balance between recall and precision. Also, C4.5+Terms+Over performs better than the baseline classifier, generalizing beyond the training examples. Furthermore, in these cases it performs better than the reasoner, which means that it locates subsumptions that cannot be inferred by using the equivalence relations produced by SEMA (results in category A7 in Fig. 5 and Fig. 6 are very depictive).

Here we have to comment about categories E1, E2, F1, and F2: Although the conceptualizations of the target ontologies differ from the source, there is a special detail that highly favors the reasoner: In F1 and E1 the concepts' hierarchy is flattened or expanded, respectively, but the initial concepts (along with their properties and restrictions) defined in the first ontology remain almost unchanged in the second ontology. This means that the reasoner can relatively easily infer subsumptions through the equivalence relations returned by SEMA, in conjunction with the subsumption relations defined in each ontology hierarchy. Furthermore, the newly introduced concepts in E1 and E2 have no defined properties at all, a fact that lowers the discriminating ability of *CSR* when it uses properties as features. The same applies to categories F2 and E2, but now the lexical information is suppressed in the second ontology.

In Fig. 9 we present the number of relations computed, which, contrary to what CSR assesses, are equivalence rather than subsumption relations. As it is shown, C4.5 which is the best performing classifier has an average of one or less in all types of ex-

periments. This is a really important feature of *CSR*, as it can perform a "filtering" in the results of any mapping system that locates equivalence relations [2].

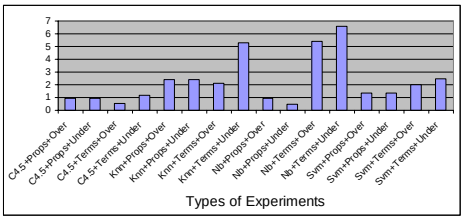

Fig. 9. Confused Equivalencies of *CSR*.

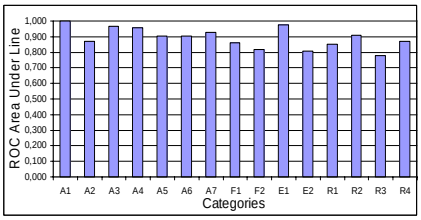

Fig. 10. ROC areas under line values.

To further assess the quality of the classifier with the best overall results, we performed ROC analysis (Fig. 10). In our case, ROC analysis indicates the "goodness" of the classifier in classifying testing examples in the distinct classes "$\sqsubseteq$" and "*R*". It is generally accepted that values ranging in [0.5, 0.6] indicate a failure in the classification task, values ranging in [0.6, 0.7] indicate a poor classifier, values ranging in [0.7, 0.8] indicate a fair classifier, values ranging in [0.8, 0.9] indicate a good classifier and finally values in [0.9, 1.0] indicate an excellent classifier.

By examining the ROC area under line values of the *CSR* method with C4.5 in all types of experiments, it is obvious that the classifier is always "good" and in the majority of the test cases (8/15) can be characterized as "excellent". It must be stated that these values depict that, although the performance of the classifier in the class "*R*" is of no evident interest for the ontology alignment problem, as in these cases the classifier cannot decide, the *CSR* method performs even better there.

6 Conclusions and Future Work

In this paper we propose the *CSR* method. *CSR* aims to the computation of subsumption relations between concept pairs of two distinct ontologies by exploiting properties' equivalence mappings, as well as appearances of terms in concepts' vicinity. Towards this goal, *CSR* assesses whether concept pairs of the source ontologies belong to the subsumption relation by means of a classification task using state of the art machine learning methods. Given a pair of concepts from two ontologies, the objective of *CSR* is to identify patterns of concepts' features (properties or terms) that provide evidence for the subsumption relation among these concepts. For the learning of the classifiers, the proposed method generates training datasets from the source ontologies specifications, tackling also the problem of imbalanced training datasets.

Experimental results show the potential of the method: *CSR* generalizes effectively over the training examples, showing (a) the importance of both properties and terms to assessing the subsumption relation between concepts of discrete ontologies (b) the

importance of incorporating more precise property mapping methods into the process, (c) the potential to further improve the method via the incorporation of more types of features, via the combination of different types of features or via its combination with other methods.

Lastly, it must be pointed that CSR manages to discriminate effectively between equivalence and subsumption relations. This is a really important feature of CSR, as it can be used for filtering equivalences computed by other alignment methods [2].

Acknowledgments. This research project is co-financed by E.U.-European Social Fund (75%) and the Greek Ministry of Development-GSRT (25%).

References

1. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. Journal on Data Semantics IV, LNCS, vol. 3730, pp. 14--171 (2005)
2. Svab, O., Svatek, V., Stuckenschmidt, H.: A Study in Empirical and 'Casuistic' Analysis of Ontology Mapping Results. In: ESWC, Innsbruck, Austria (2007)
3. Mitchell, T.: Machine Learning. The McGraw-Hill Companies, Inc. (1997)
4. Giunchiglia, F., Yatskevich, M., Shvaiko, P.: Semantic Matching: Algorithms and implementation. Journal on Data Semantics, IX (2007)
5. Bouquet, P., Serafini, L., Zanobini, S., and Sceffer, S. 2006: Bootstrapping semantics on the web: meaning elicitation from schemas. In: WWW, Edinburgh, Scotland (2006)
6. Aleksovski, Z., Klein, M., Kate, W, Harmelen F.: Matching Unstructured Vocabularies Using a Background Ontology. In: EKAW, Podebrady, Czech Republic (2006)
7. Gracia, J., Lopez, V., D'Aquin, M., Sabou, M, Motta, E., Mena, E.: Solving Semantic Ambiguity to Improve Semantic Web based Ontology Matching, In: Ontology Matching Workshop, Busan, Korea (2007)
8. Risto G., Zharko A., Warner K.: Using Google Distance to weight approximate ontology matches. In: WWW, Banff, Alberta, Canada (2007)
9. Hage, W.R. Van, Katrenko, S., Schreiber, A.Th.: A Method to Combine Linguistic Ontology Mapping Techniques, In: ISWC, Osaka, Japan (2005)
10. Cimiano P., Staab, S.: Learning by googling, In: SIGKDD Explor. Newsl., USA (2004)
11. Jerome D., Fabrice G., Regis G., Henri B.: An interactive, asymmetric and extensional method for matching conceptual hierarchies. In: EMOI – INTEROP Workshop, Luxembourg (2006)
12. Chih-Chung Chang, Chih-Jen Lin: LIBSVM : a library for support vector machines (2001)
13. Spiliopoulos, V., Valarakos, A.G., Vouros, G.A., Karkaletsis, V.: SEMA: Results for the ontology alignment contest OAEI 2007. In: Ontology Matching Workshop, OAEI, Busan, Korea (2007)
14. Ontology Alignment Evaluation Initiative, http://oaei.ontologymatching.org/
15. Japkowicz, N.: The Class Imbalance Problem: Significance and Strategies. In: ICAI, Special Track on Inductive Learning, Las Vegas, Nevada (2000)