Web Semantics: Science, Services and Agents on the World Wide Web xxx (2010) xxx–xxx

Contents lists available at ScienceDirect

Web Semantics: Science, Services and Agents on the World Wide Web

journal homepage: www.elsevier.com/locate/websem

On the discovery of subsumption relations for the alignment of ontologies

Vassilis Spiliopoulos [a,*], George A. Vouros [a,*], Vangelis Karkaletsis [b]

[a] AI Lab, Information and Communication Systems Engineering Department, University of the Aegean, Karlovassi, Samos 83 200, Greece
[b] Institution of Informatics and Telecommunications, NCSR "Demokritos", Athens, Greece

ARTICLE INFO

Article history:
Received 15 March 2009
Received in revised form
22 December 2009
Accepted 4 January 2010
Available online xxx

Keywords:
Ontology alignment
Subsumption
Supervised machine learning

ABSTRACT

For the effective alignment of ontologies, the subsumption mappings between the elements of the source and target ontologies play a crucial role, as much as equivalence mappings do. This paper presents the "Classification-Based Learning of Subsumption Relations" (*CSR*) method for the alignment of ontologies. Given a pair of two ontologies, the objective of *CSR* is to learn patterns of features that provide evidence for the subsumption relation among concepts, and thus, decide whether a pair of concepts from these ontologies is related via a subsumption relation. 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 of both artificially created and real world cases, and discusses the potential of the method.

1. Introduction

Despite 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 mappings between the elements of ontologies, in this paper we conjecture that this is not enough: to deal effectively with the ontologies' alignment problem, we also have to deal with the discovery of non-equivalence mappings among ontology elements. To this end, in this work we investigate the discovery of subsumption mappings. Although the usefulness of subsumption mappings may be known to the ontology alignment community, to the best of own knowledge, no alignment method has thoroughly investigated the computation of such mappings. Therefore, the progress that has been made towards the location of subsumption mappings is not sufficient, in comparison to the progress made to the computation of equivalence mapping relations.

Subsumption mappings are particularly useful when we deal with ontologies whose conceptualizations are at different "granularity levels": in these cases, the elements (concepts or properties) of an ontology are more generic than the corresponding elements of another ontology. Although subsumption mappings between the elements of two ontologies may be deduced by exploiting equivalence mappings between other elements (e.g. a concept C_1 is subsumed by all subsumers of C_2, if C_1 is equivalent to C_2), in the extreme cases where no equivalence mappings exist, or in cases where the assessed/provided equivalences are erroneous, this cannot be done effectively. This paper conjectures that the direct 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, since, 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 the 1st ontology (which is equivalent to the concept `Reference` in the 2nd ontology), and `Publication` in the 2nd ontology (which is equivalent to the concept `Work` in the 1st ontology). Each of these ontologies elaborate on the specification of different concepts: the second ontology elaborates on the concept `Publication`, defining different kinds of publications, while the first ontology elaborates on the concept `Citation`, defining different kinds of citations. Given these ontologies, the fact that equivalent concepts 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.

In case that the equivalence mappings between the concepts of the two ontologies are not known, conclusions concerning subsumption mappings between the concepts of the two ontologies cannot be drawn by a reasoning mechanism. This case shows in a very clear way the necessity to discover equivalence and sub-

* Corresponding authors. Tel.: +30 6937363087; fax: +30 2273082229.
E-mail addresses: vspiliop@aegean.gr (V. Spiliopoulos), georgev@aegean.gr (G.A. Vouros), vangelis@iit.demokritos.gr (V. Karkaletsis).

1570-8268/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.websem.2010.01.001

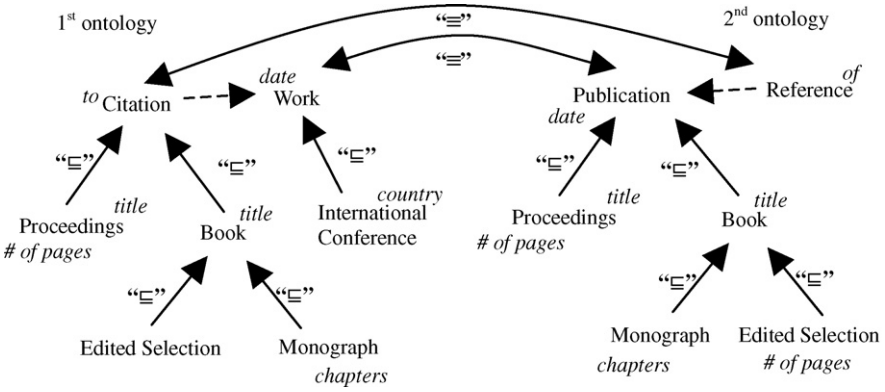

Fig. 1. Example ontologies for assessing the subsumption relation between concepts.

sumption relations between the concepts of the source and target ontologies.

In the second case where equivalence mappings between the concepts of the two ontologies are known, these can be exploited by a reasoning mechanism to deduce subsumption mappings. However, in case that equivalence mappings have been computed by an alignment mechanism, then wrong equivalences shall provide evidence to wrong subsumption mappings. For example, a state of the art alignment tool may wrongly assess that the concept `Monograph` in the 1st ontology is equivalent to the concept `Monograph` in the 2nd ontology, as their ontological features (e.g. labels, defined properties, direct super/sub concept, and depth in the taxonomy) are exactly the same, as far as their surface appearance is concerned. A reasoning mechanism exploiting this equivalence relation would wrongly deduce that the concept `Monograph` in the 1st ontology is subsumed by the concepts `Book` and `Publication` in the 2nd ontology. However, the correct relation is that the concept `Monograph` in the 1st ontology is subsumed by the concept `Reference` in the 2nd ontology.

Furthermore, even if one (human or software entity) can assess that the concept `Work` in the 1st ontology is equivalent to the concept `Publication` in the 2nd ontology, a reasoning mechanism exploiting this knowledge would correctly infer that `International Conference` is subsumed by `Publication`, but it would not be able to place `International Conference` under its direct subsumer (i.e. its correct place in the hierarchy), which in this example is the concept `Proceedings` (this is so since the concept `International Conference` represents publications that appear in the proceedings of international conferences, while `Proceedings` represent publications that appear in any kind of scientific event, e.g. workshops). This example shows that even if we exploit correct equivalences to derive subsumptions, there are cases where the subsumptions found are not sufficient for the merging of the involved ontologies.

The above examples provide evidence towards our conjecture: What is clearly needed is a method that shall discover subsumption relations between concept pairs of two distinct ontologies, separately from subsumptions and equivalences that can be deduced by a reasoning mechanism. In other words, the method should directly pursue the location of subsumption mappings, without necessarily relying on equivalence mappings.

This paper deals with the problem of discovering subsumption mappings between concepts of two distinct ontologies, without relying on known equivalence mappings among them. This is done by using the "Classification-Based Learning of Subsumption Relations" (*CSR*) method for the alignment of ontologies. *CSR* computes subsumption mappings between concept pairs of two ontologies by means of a classification task, using state of the art supervised machine learning methods. Specifically, given a pair of concepts from the source and target ontologies, the classification method "locates" a hypothesis concerning relation of concepts, which best fits to the training examples [3], while generalizing beyond them. The training examples are generated by exploiting both the source and target ontologies, without requiring human intervention (this is thoroughly explained in Section 4). The classification mechanisms proposed exploit features of concepts of different types, for the representation of concept pairs. A detailed description of the classification features used is provided in Section 4.

The basic version of *CSR* has been presented in [4]. The work presented in this article extends the one presented in [4] to the following: (a) we investigate six more different types of classification features (there are two such types in [4]), that improved the efficiency of the method in terms of precision and recall (b) we introduce a new dataset balancing technique based on the semantics of the source and target ontologies, again with a positive impact on the method, and finally (c) we provide a thorough evaluation of *CSR* using three different datasets (in contrast to the one used in [4]).

The machine learning approach has been chosen since (a) there are no evident generic rules that capture *directly* the existence of a subsumption relation between a pair of ontology elements (e.g. by means of their surface appearance, labels/vicinity similarity or dissimilarity), and (b) concept pairs of the same ontology can provide examples for the subsumption relation, making the method self-adapting to the idiosyncrasies of specific domains and conceptualizations provided, and non-dependant to external resources.

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 necessary background knowledge concerning supervised machine learning and the classification methods used. Furthermore, this section provides background information concerning probabilistic topic models, which are used for the generation of classification features. Section 4 presents the proposed classification-based method for the discovery of subsumption mappings, and discusses specific choices regarding method's alternative configurations. Section 5 presents and thoroughly discusses the experimental settings, as well as the results. Finally, Section 6 concludes the paper by pointing out the main aspects of our method and sketching further work for its enhancement and exploitation.

2. Problem definition and related work

2.1. Problem definition

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

Please cite this article in press as: V. Spiliopoulos, et al., On the discovery of subsumption relations for the alignment of ontologies, Web Semantics: Sci. Serv. Agents World Wide Web (2010), doi:10.1016/j.websem.2010.01.001

the intended meaning of the terms included in the signature [5,6]. In other words, A includes the formal definitions of ontology elements that are lexicalized by 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 words lexicalizing ontology properties and ontology concepts, respectively. Let also W be the set of distinct words that are in S, or that are extracted from labels, comments or instances of ontology elements. For example, concerning the 1st ontology depicted in Fig. 1, a fragment of its representation is as follows: $S_p = \{$`to`, `date`, `# of pages`, `title`, ...$\}$, $S_c = \{$`Citation`, `Work`, `Proceedings`, ...$\}$, $A = \{$`Proceedings` $\sqsubseteq$ `Citation`, ...$\}$, and $W = \{$`to`, `Citation`, `date`, ...$\}$.

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 such that $A_2 \models f(A_1)$, i.e. all interpretations that satisfy O_2s axioms also satisfy O_1's translated axioms. However, instead of a function that specifies equivalences among ontology elements, we may align ontologies by articulating five different kinds of binary relations between the elements of the source and target ontologies: Namely, equivalence ($\equiv$), subsumption (inclusion) ($\sqsubseteq$ or $\sqsupseteq$), mismatch ($\perp$) and overlapping ($\sqcap$). In this case, the ontology alignment problem can be stated as follows: Classify any pair $(\mathtt{C}^1, \mathtt{C}^2)$ of elements of the source and target ontologies, such that $\mathtt{C}^i$ is a term in S_i, $i = 1,2$, to the above relations, consistently w.r.t. to the semantics of specifications in the source and target ontologies, and to the computed relations. Having classified any pair $(\mathtt{C}^1, \mathtt{C}^2)$ of elements to these relations, ontologies O_1 and O_2 can be merged, resulting to a new consistent and coherent ontology. For example, concerning the ontologies depicted in Fig. 1, such pairs are the following: (a) (`Citation, Reference`) classified to the equivalence relation ($\equiv$) and (b) (`Proceedings, Reference`) classified to the subsumption relation ($\sqsubseteq$).

In this paper we deal with the problem of computing subsumption mappings (*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 $W_1 \cup W_2$ of distinct words that appear in both ontologies, and *optionally* (c) a morphism $f: S_1 \rightarrow S_2$ from the lexicalizations of properties or concepts of the source ontology to the lexicalizations of the properties or concepts of the target ontology (specifying equivalence mappings of properties and concepts, respectively), classify each pair $(\mathtt{C}^1, \mathtt{C}^2)$ of concepts, where $\mathtt{C}^1$ is a term in S_{1c} and $\mathtt{C}^2$ is a term in S_{2c}, to two distinct classes: To the "subsumption" ($\sqsubset$) class (meaning that $\mathtt{C}^1 \sqsubset \mathtt{C}^2$), or to the class "$\lambda$". 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 relations. We have to emphasize that in this paper we aim to compute strict subsumption relations among classes. Therefore, equivalent ontology classes must not be classified to the "subsumption" but to the "λ" class.

Although the proposed method is ontology-language neutral (i.e. different implementations of the proposed method, can handle different ontology languages), throughout this paper we assume that ontologies are specified in the OWL-DL language [7].

The OWL-DL language has been given its name due to its background on Description Logics (DLs) [7] and specifically to the SHOIN(D) description language [8]. The elementary descriptors of DLs are *atomic concepts* (concepts) and *atomic roles* (properties). Descriptions (i.e. terminological axioms) can be built from these descriptors inductively, using a set of *constructors*.

The formal semantics are defined using the notion of *interpretation* I, that consists of a non-empty set Δ^I (the domain of the interpretation) and an interpretation function, which assigns to every atomic concept `A` a set $\mathtt{A}^I \subseteq \Delta^I$ and to every atomic role R a binary relation $\mathtt{R}^I \subseteq \Delta^I \times \Delta^I$. Moreover, an interpretation I *satisfies* a strict subsumption relation $\mathtt{C} \sqsubset \mathtt{D}$ iff $\mathtt{C}^I \subset \mathtt{D}^I$. If A is a set of axioms, then I satisfies A, iff I satisfies each element in A and thus constitutes a *model* of A. Concerning the strict subsumption relation that we are interested in this paper we specify that a concept `C` is subsumed by a concept `D` (i.e. $\mathtt{C} \sqsubset \mathtt{D}$) with respect to A (i.e. a set of axioms describing `C` and `D`), if $\mathtt{C}^I \subset \mathtt{D}^I$ for every model I of A.

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 mappings between ontology elements [1,5] (e.g. concepts and properties). As a result, there has been a dramatic increase in the efficacy and efficiency of the methods that locate equivalences among ontology elements (i.e. equivalence mappings), while subsumption mappings have not been thoroughly investigated. In the next paragraphs we present the alignment methods that are mostly related to our work and also target the location of subsumption mappings.

The Semantic Matching approach [9] implemented by the S-Match system, deals with the computation of equivalence, subsumption, intersection (overlapping) and disjoint mappings between concepts of ontologies. The mapping relation is computed by (a) expressing the input ontologies' concepts into propositional formulas and (b) by transforming the problem of aligning ontologies into a propositional satisfiability problem. For the expression of concepts as propositional formulas the method exploits: (a) the labels of concepts, (b) the structural knowledge of the ontology, (c) semantic knowledge extracted from WordNet senses, and (d) mappings among concepts, which are computed utilizing a set of methods, such as string based methods, n-gram based methods and WordNet-based methods. Recently [9], the Semantic Matching method exploits ontology properties, by expressing the problem as a Description Logics reasoning problem (instead of a propositional satisfiability one).

Another interesting approach that aims to the discovery of subsumption mapping relations is presented in [10]. The authors introduce the Wordnet Description Logics (WDL) language as a way to bridge the semantic gap between two different ontologies and apply Description Logics reasoning services to the two ontologies, as if they were a single one. The authors argue that primitives (concepts and properties) of any DL language do not have an "intended" meaning and for this reason they propose the "grounding" of their interpretation to WordNet senses that best represent their intended meaning. Although the authors do not focus on the process of how senses are mapped to ontology elements, they provide specific rules that translate the input ontologies into WDLs formulas, in order to infer relations among elements of the input ontologies.

Similarly, in [11] the authors propose the exploitation of background knowledge in the form of domain ontology for bridging the semantic gap between two different ontologies. Specifically, elements of the two input ontologies are mapped through equivalence relations (called anchoring matches) to elements of a domain ontology (called anchors). Then, based on the relationships among the anchors encoded in the domain ontology, the method infers how

[1] This means that a pair of concepts belonging to "λ" may belong to the strict subsumption relation, as class "λ" states our ignorance whether the subsumption relation holds for a pair of concepts. This is due to the open world semantics of the OWL-DL language.

Please cite this article in press as: V. Spiliopoulos, et al., On the discovery of subsumption relations for the alignment of ontologies, Web Semantics: Sci. Serv. Agents World Wide Web (2010), doi:10.1016/j.websem.2010.01.001

the elements of the source ontology are related to the elements of the target ontology.

In [12] the previous approach is further developed, as authors argue that the main reason for producing erroneous mappings is the wrongly assessed anchoring matches. As a result, they propose the use of sense disambiguation techniques via the PowerAqua tool [13], and focus on (a) improving the quality of the automatically generated anchoring matches, and (b) discarding erroneous mappings. In more detail, the disambiguation technique is based on the *synonymy degree* measure provided by the PowerAqua tool, in conjunction with a WordNet based technique proposed in their article. The main intuition of the WordNet based technique is that the similarity of two concepts depends on the relations among the WordNet senses that best describe their intended meaning.

The authors in [14], instead of targeting to the problem of locating subsumption mappings between concepts, they argue that in ill-defined domains such as internet music taxonomies, it is of paramount importance to loosen the formal constraints of the subsumption relation. Specifically, the authors define a measure (named *sloppiness*) that shows whether it is possible for a concept A to be more general than a concept B, without being more general from all subconcepts of B. For defining sloppiness, the authors divide the subsumption checking problem to a number of subproblems, where sloppiness expresses the number of sub-problems that can not be addressed successfully. The impact a sub-problem has on the main subsumption problem is encoded with a subproblem specific weight. The computation of these weights is made by exploiting the *normalised Google distance*, which captures the probability of two terms (labels of concepts in this case) to coappear in a web page.

Two more Google-based approaches aim to the location of subsumption mappings [15,16] by exploiting Hearst patterns [17]. The validity of subsumptions is tested by exploiting the hits returned by the Google search engine. For example, among other techniques, the authors propose the use of a threshold value that indicates whether the hits are sufficient to conclude that the tested subsumption relation holds.

Another approach for the location of subsumption mappings, characterized by the usage of heuristic rules, is the one introduced by the TaxoMap system [18]. Concerning the discovery of subsumption mappings, TaxoMap uses two rules, the intuition of which be summarized as follows: (a) if a concept has a label that is included in the label of another concept, then the first concept is subsumed by the second, and (b) if there are three concepts from one ontology that are the most similar ones to a single concept A from the other ontology, and these three concepts share a common subsumer, then the concept A is subsumed by their common subsumer.

Although machine learning techniques have been used in several works for schema and ontology mapping [1,5], researchers aim mostly to the discovery of equivalence mappings between ontology elements and do not focus on the computation of subsumption mappings between ontology elements, as we do in this work. To the best of our knowledge, the only approach that targets to the computation of subsumption mappings is presented in [19]. Specifically, the authors propose a method based on the Implication Intensity theory. The main intuition of the approach is that one concept is more specific than another, if the words that appear in the documents associated to the first concept tend to be a sub-set of the words that appear in the documents associated to the other one. The method takes as input a hierarchy of concepts and a set of documents, each one being classified under a specific concept. Then, the proposed method is applied in order to locate strong derivations between sets of terms that appear in the documents, and as a consequence between their indexing concepts.

Another method similar to the one we propose is detailed in [20]. Specifically, the authors propose a method, called oPLMap, for automatically locating mappings among web directories (i.e. hierarchy of classes with associated documents to each class). oPLMap is based on a logical framework, which is combined with probability theory (probabilistic Datalog), and aims at finding the optimum mapping (i.e. the mapping with the highest matching probability). In terms of the oPLMap method, a mapping determines a "similarity" relationship between classes of two web directories. For example, a mapping may state that an instance of a class A from the source web directory is also an instance of a class B from target web directory. These facts may imply either a subsumption or an equivalence mapping between classes. This is the major difference of this work and the one proposed in this paper: *CSR* computes strict subsumption relations among ontology classes.

All the aforementioned approaches for the computation of subsumption mappings, except TaxoMap, have strong dependence on external resources: WordNet, domain ontologies or text corpora. The method proposed in this paper has been devised to be as generic as possible and independent of any external resource, devoting special attention to the idiosyncrasies of the ontologies considered. Specifically, in this paper we consider the subsumption computation problem as a classification problem, where a classifier has to assess whether a pair of concepts belongs to the subsumption relation. The source and target ontologies are exploited in order the method to generate the appropriate examples for the training of the classifier. This is of great advantage, since (a) the proposed method depends only from the source and target ontologies and is independent from any third/external domain resource (lexicon, thesaurus or text corpora), and (b) the proposed method tunes itself to the idiosyncrasies of the input ontologies.

3. Background knowledge

3.1. Supervised classification

Classification is one of the main problems addressed within the machine learning discipline. It concerns the classification of example cases into a discrete set of classes. When the number of classes is restricted to two, the problem is referred to as a binary classification problem.

In supervised classification the inducer is fed with training examples (data set) $E = \{E_1, E_2, \ldots, E_m\}$. Each training example $E_j \in E$ is associated with a label which indicates the class it belongs to. More formally, each training example is a tuple $E_j = (\vec{x}_j, y_j)$, where $\vec{x}_j \in R^n$ is a vector of features' values (feature vector) of the training example sampled from a distribution D, and $y_j \in Y$ is the class to which $\vec{x}_j$ belongs (Y is the set of classes). The objective of the supervised classification is to induce an unknown function $c: R^n \rightarrow Y$ (classifier), that maps previously unseen instances $\vec{x}$, sampled from the same distribution D, to values in Y. The ith component of the vector $\vec{x}_j$ is termed the *feature* i of $\vec{x}_j$.

As already stated in Section 2.1, the subsumption computation problem can be defined as a binary classification problem, with two defined classes: class "subsumption" ($\sqsubset$) and class "λ", i.e. $Y = \{\sqsubset, \lambda\}$.

3.2. Classifiers

In the context of studying the subsumption computation problem, we have used specific implementations of the following well studied and most popular types of classifiers:

(i) *Probabilistic classifiers* specify the function c as a probabilistic function, assessing the probability $p(x_j, y_j)$ that x_j falls within a category y_j. From this category we have selected the archetypical Naïve Bayes (Nb) classifier [21]. Naïve Bayes is based on the application of the Bayes theorem and is trained in a

Please cite this article in press as: V. Spiliopoulos, et al., On the discovery of subsumption relations for the alignment of ontologies, Web Semantics: Sci. Serv. Agents World Wide Web (2010), doi:10.1016/j.websem.2010.01.001

supervised learning setting using maximum likelihood for estimating the parameters of its model. The Naïve Bayes classifier is based on the assumption that predictor variables (i.e. features) are independent random variables. Although such an assumption is quite strong (it does apply very rarely), Naïve Bayes performs surprisingly well in some real-world problems (e.g. spam filtering). In the subsumption computation problem, this assumption concerns the independence of features (these are presented in Section 4) in a pair of concepts. However, this is not the case in our problem: For instance, if the concept `Proceedings` is related to the property `conference`, the probability of being related to the property `hasPet` is affected.

(ii) *Memory-based classifiers* [22] (sometimes called "lazy" 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. The *k*-nearest neighbor (knn) is the most popular classifier of this method, where *k* defines the number of instances (neighbors) retrieved from memory and used for predicting the class of an instance. Usually, the Euclidean distance is been used for measuring the distance between two instances, while the importance of each neighbor to determine the class of an instance is inversely proportional to its distance from the instance.

The biggest advantage of knn is its simplicity. On the other hand, its prediction accuracy degrades as the number of features grow. Furthermore, it is computationally expensive when it is applied to large corpora.

(iii) *Support Vector Machines* (*SVMs*) *based classifiers* [23]. Support vector machines map feature vectors to a higher dimensional space where a maximal separating hyperplane is constructed. Although this hyperplane separates the instances belonging to different classes linearly, in the initial dimensional space the instances may be non-linearly separable. The transformation of the data to the new space is made through functions called kernels. Commonly used kernel functions are the Polynomial, the Radial Basis and the Gaussian Radial Basis functions. In the binary classification problem two parallel hyperplanes are been constructed, categorizing the feature vectors. The separating hyperplane is the one that maximizes the distance between the two parallel hyperplanes, since the larger the margin or distance between the parallel hyperplanes is, the better the generalization error of the classifier will be.

The major advantage of SVMs is that they are quite effective as non-linear classifiers. However, this comes together with their biggest disadvantage: The need for fine tuning various parameters (e.g. the kernel determination and regularization coefficient). Indeed, different settings greatly influence the results. In other words, SVMs do not often work "out of the box". Finally, these classifiers are computationally expensive.

(iv) *Decision tree classifiers* [24]. Decision tree classifiers exploit a tree structure in which each interior node corresponds to a feature. The branch from a node to a child (arc) represents a possible value of that feature, and a leaf node represents a possible classification class. Decision trees are trained by splitting the training examples into subsets based on a feature value test. This process is recursively applied to the resulting subset, until a subset cannot be split any further, or a singular classification can be applied to the examples of the subset. One of the most commonly used decision tree classifier is the C4.5. C4.5 utilizes the normalized Information Gain measure in order to choose the feature that best splits the data.

Decision tree classifiers have several advantages [24] that can be proved very helpful in the subsumption computation problem. This happens because: (a) they perform well when applied to large amounts of data, without being computationally expensive. In our experiments there are cases where the number of features is very high (>20,000 features) and there is a high number of training examples, (b) 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 "$\sqsubset$,". (c) Decision trees are tolerant to errors in the training set. This is true as far as the training examples, as well as the features values, are concerned. Moreover, (d) decision tree-based methods are non-linear classification methods, which means that they are able to perform well even if the training examples are overlapping (this is discussed in the next section). Last but not least, (e) decision trees tend to perform well "out of the box", without the need of tuning any parameters.

3.3. Learning from imbalanced data sets

It is very important for the efficiency of any classifier that the training dataset is balanced in numbers. In other words, the training examples of all classes should be equal in numbers [25]. To understand the problem, let us consider the training examples depicted in Fig. 2. The majority class (i.e. the class with the highest number of training examples) is represented with "−", while the minority (i.e. the class with the fewer training examples) with "+". In order to be able to represent the training examples, we assume that the feature vectors are of length equal to two. In Fig. 2(a) there is a high degree of imbalance between the two classes (there are much more "−" than "+"). This, in conjunction to the fact that classes are non-linearly separable, makes the classification task quite difficult. To illustrate the difficultly, it is obvious that an 1-nearest neighbour classifier would wrongly classify many "+" instances, as their first most nearest neighbour will most probably be a "−" instance. A similar behaviour can be observed for all classifiers presented in the previous subsection. In contrast to that, Fig. 2(b) depicts an "easier" classification scenario (linear separable, nearly balanced classes).

In the literature there are numerous works that deal with the data set imbalance problem [26]. In the context of the subsumption computation problem, to deal with the imbalanced data sets, we examine three alternatives:

(i) *Random over-sampling*. This strategy randomly selects examples from the minority class (i.e. the one with fewer cases) and re-adds them in this class, until the two classes are equal in numbers. A common belief is that random over-sampling can increase the likelihood of overfitting, since it produces exact copies of examples [26]. On the other hand, there are various works [25,26] stating that random over-sampling performs effectively, especially when it is applied in combination with decision tree-based classifiers.

(ii) *A variation of random under-sampling*. This strategy selects to discard a subset of the training examples of the majority class, so that the majority and the minority classes to contain the same number of examples. The major drawback of any under-sampling method is the possibility of discarding potentially useful training examples that could be important for the classi-

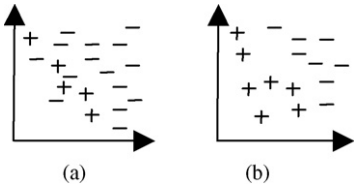

Fig. 2. Distribution of training examples in a 2-dimensional space.

Please cite this article in press as: V. Spiliopoulos, et al., On the discovery of subsumption relations for the alignment of ontologies, Web Semantics: Sci. Serv. Agents World Wide Web (2010), doi:10.1016/j.websem.2010.01.001

fier [26]. For this reason, the proposed under-sampling method tries to avoid such a situation, by exploiting the semantics of the input ontologies. A detailed description of the method is provided in Section 4.

(iii) *Artificial training examples*. This is a method proposed here. It exploits the semantics of the input ontologies in order to generate artificial examples of the minority class, for balancing the dataset. Although this method overcomes the problem that derives from the application of random over-sampling, there is always the chance that an artificially created example is not representative of its class and as a result it provides "noise". The proposed method tries to avoid such a situation, by exploiting the semantics of the input ontologies. A detailed description of the method is provided in Section 4.

3.4. Probabilistic topic models

In conjunction to the other types of classification features we also study the efficiency of statistically generated features (called latent features/variables). Latent variables were introduced in the probability topic models [27]. A latent variable is a probability distribution over words, and a probabilistic topic model specifies a certain *generative process*: Documents (assumed to be "bag of words") can be generated by mixtures of *latent variables*. By emphasizing on latent variables rather than words, probabilistic topic models aim to capture the "significant" features in terms of which different elements (documents) can be represented.

Fig. 3(a) depicts an instance of the generative process: Given (a) two latent variables specifying the probability of each word (shown inside the parenthesis next to each word), (b) the probability according to which each variable contributes to the generation of each document (shown by the arrows and the numbers labeling them), three different documents have been generated, emphasizing on topics, whose mixture is represented by a specific combination of the latent variables. As it is depicted in Fig. 3(a), the *generative process* makes no assumption about the exclusive assignment of a word to a variable, thus capturing the notion of polysemy. Synonymy relations can also be incorporated in the process, as different words with similar meanings may be assigned to the same latent variable. These are important, considering that such phenomena occur frequently as far as the lexicalization of ontology elements is concerned.

While the above process concerns the generation of documents by known mixtures of known latent variables, we are interested in the reverse process (depicted in Fig. 3(b)): Given documents that express the meaning of ontology elements, we need to infer the latent variables along with their mixture proportions for each document. Given that each such document corresponds to an ontology element, ontology elements are finally been represented by means of these latent features. In Section 4, we present in detail how latent features are being used for the representation of training and testing examples.

3.5. Latent dirichlet allocation (LDA)

Every probabilistic topic model assumes a specific generative process for a document. This assumption is necessary in order to be able to reverse the process and infer the latent variables and their mixture proportion for each document, as explained in the previous section.

To introduce some notation, $P(z_i = j)$ stands for the probability that the jth latent feature was sampled for the ith word, and $P(w_i|z_i = j)$ stands for the probability of the occurrence of word w_i given the latent feature j. To simplify the notation, $P(z)$ (these are the numbers labeling the arrows in Fig. 3) and $P(w|z)$ (the probabilities of the words in latent variables in Fig. 3) indicate which latent features can be used for expressing the content of a particular document and which words are "important" for each latent feature, respectively. Moreover, Poisson (ξ), Dirichlet (α) and Multinomial (ϑ) stand for the corresponding well known probability distributions along with their parameters.

The latent dirichlet allocation model [28], given a predefined number of latent features T, assumes the following generative process for each document:

1. Choose $N \sim$ Poisson(ξ).
2. Choose $\vartheta \sim$ Dirichlet(α), $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_T)$
3. For each of the N words in the document (i refers to the ith word):

- Choose a latent feature $z_i \sim$ multinomial (ϑ).
- Choose a word w_i from $P(w_i|z_i)$, a multinomial probability distribution conditioned on the latent feature z_i.

Poisson is introduced for modeling a realistic assumption of the document length distributions, as it expresses the probability of a number of words to appear in a document of length N, if words appear with an average rate (indicated by parameter ξ). Each α_j, $j \in [1,T]$ entry of a can be interpreted as a prior observation count for the number of times topic j is sampled in a document.

Having said that, the model specifies the following distribution over words within a document:

$$P(w_i) = \sum_{j=1}^{T} P\left(w_i \left| z_i = j\right.\right) P(z_i = j)$$

In the reverse process, where documents are known, standard machine learning/statistical techniques can be used to infer the parameters $P(z)$ and $P(w|z)$ according to which the known documents have been generated. For this purpose we are using a specific Markov Chain Monte Carlo (MCMC) process called Gibbs sampling.[2] It should be stated that the reverse process does not infer the number of latent features T. T is a parameter of the process and its value influences the inference of the parameters $P(z)$ and $P(w|z)$. The interested reader is referred to [27] for a detailed explanation of this process.

4. The *CSR* method

As already pointed, *CSR* [4] addresses the subsumption computation problem as a binary classification task, using state of the art supervised machine learning methods. The discrete steps of the *CSR* method, as depicted in Fig. 4, are the following:

(i) Enhancement of ontology hierarchies: Reasoning services are being used for inferring all subsumption relations in each ontology [7]. This is a necessary step as it affects the generation of the training dataset specified in Section 4.2.

(ii) Generation of features for the classifier: In previous lines of our work [4] *CSR* exploited two different types of classification features based on properties of concepts, and on words appearing in the vicinity of concepts. In this paper we investigate eight different types of classification features. This is further detailed in the next subsection.

(iii) Generation of training examples: The sets of training examples are being generated according to the rules defined in Section

[2] The LDA model implementation with Gibbs sampling that we have used is in http://www.arbylon.net/projects.

Please cite this article in press as: V. Spiliopoulos, et al., On the discovery of subsumption relations for the alignment of ontologies, Web Semantics: Sci. Serv. Agents World Wide Web (2010), doi:10.1016/j.websem.2010.01.001

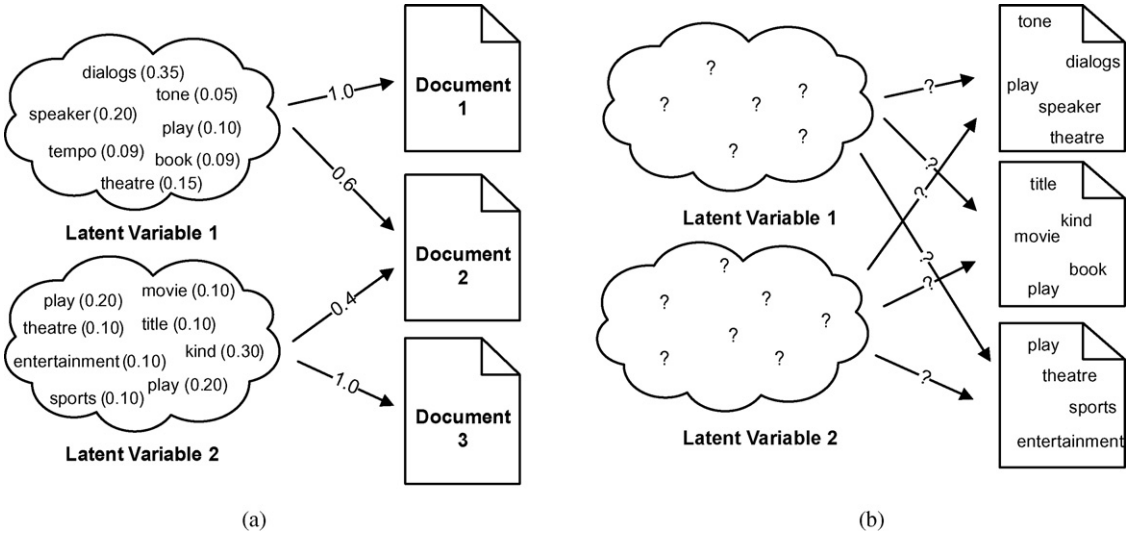

Fig. 3. (a) The generative process and (b) the inference process (reverse generative process).

4.2. The balancing of the training dataset is an important issue that is being tackled in this step, as well.

(iv) Train classifier: The classifier is being trained using the training data set.

(v) Generation of testing pairs: Concept pairs are being classified by the trained classifier. The space of possible concept pairs is "pruned" according to the method presented in Section 4.4.

Furthermore, although this is not necessary, we have configured *CSR* to exploit equivalence mappings between elements. Equivalence mappings are being computed automatically by the SEMA mapping tool [29]. These equivalences may concern properties or concepts. As it will be detailed in Section 4.1, when equivalences between properties of concepts are being computed, then equivalent properties correspond to the same classification feature, allowing for more "informed" decisions to be taken concerning the classification of concept pairs. Additionally, equivalences among concepts are exploited by *CSR* to generate more training examples for the class "$\sqsubset$,". The generation of training examples is described in Section 4.2. At this point we need to recall that the exploitation of mappings of concepts does not guarantee the location of all necessary subsumptions for the merging of the input ontologies.

The main purpose of SEMA [29] is to locate equivalence mappings between the elements (i.e., concepts and properties) of the source and target ontologies and its use in the context of *CSR* is only optional. SEMA combines lexical, semantic and structural mapping algorithms: A semantic mapping method exploiting latent dirichlet allocation model [29], requiring no external resources, in combination with the lexical mapping method COCLU (COmpression-based CLUstering) [29] and a mapping method that exploits structural features of the ontologies by means of simple rules. This combination of approaches contributes towards automating the mapping process, resulting to increased recall and precision. It must be emphasized that the aggregation of the equivalence mappings produced by the individual methods is performed through their iterative execution as described in [29].

4.1. Classification features

Both training and testing examples are pairs of concepts. Given a source ontology O_1 and a target ontology O_2, a pair of concepts (C^1, C^2), where $C^1 \in O_1$ and $C^2 \in O_2$, is represented by a vector $(f_1, f_2, \ldots, f_N)$, where each f_i, $i = 1, \ldots, N$ depends on the type of features used and N is the total number of features detected in both ontologies. Subsequently, we describe the different types of features that we have used in the experiments.

Type 1: In this case N corresponds to the total number of properties defined in both O_1 and O_2 ontologies, and p_i to the ith property. In case equivalence mappings of ontology properties have been computed, equivalent properties are treated as a single property p_i that is shared by both input ontologies. To represent the direction of the subsumption relation, given a concept pair (C^1, C^2), f_i is defined as follows:

$$f_i = \begin{cases} 0, & \text{if } p_i \text{ is associated neither to } C^1 \text{ nor to } C^2 \\ 1, & \text{if } p_i \text{ is associated only to } C^1 \\ 2, & \text{if } p_i \text{ is associated only to } C^2 \\ 3, & \text{if } p_i \text{ is associated to } C^1 \text{ and } C^2 \end{cases}$$

For example, the training pair (`Citation-`O_1`, Proceedings-`O_1) in Fig. 1 is represented by the vector [1, 2, 2, 0, 0, . . ., 0], where the first three features of the vector correspond to the properties `to` (equivalently, `of`) (equivalent properties are treated as a single feature), `title`, `# of pages`. All other features correspond to

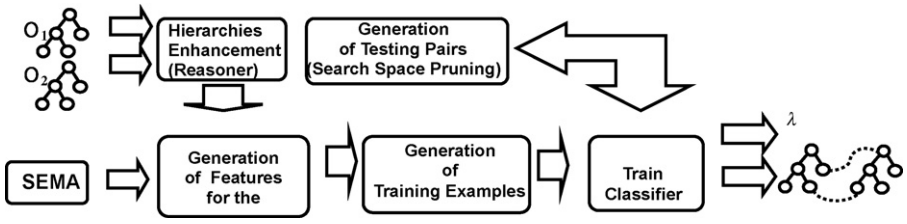

Fig. 4. Overview of the *CSR* method.

Please cite this article in press as: V. Spiliopoulos, et al., On the discovery of subsumption relations for the alignment of ontologies, Web Semantics: Sci. Serv. Agents World Wide Web (2010), doi:10.1016/j.websem.2010.01.001

properties that are not related to any of the constituent concepts of the pair and as a result their value is zero.

Using words, instead of properties, any concept C in O_1 or O_2 is represented as a vector of the form $(fr_1^j, fr_2^j, \ldots, fr_N^j)$. fr_i^j, $i = 1, \ldots, N$, $j = 1, 2$ (j indicates the left or right side of C in a pair (C^1, C^2), as shown in Fig. 5) corresponds to one of the distinct N words extracted from both ontologies O_1 and O_2, and it is equal to the frequency of this word in the vicinity of C. The vicinity of a concept comprises its local name, label or comment, its properties (including the local names, labels and comments of properties), as well as the related concepts or instances. Words are extracted from the vicinity of a concept after tokenization, stemming, and elimination of stop words.

By exploiting the equivalence, disjoint and subsumption relations between ontology elements, as well as the conjunction and disjunction constructors, we may "extend" the vicinity of an element by including words occurring: (i) in the vicinity of all of its equivalent and direct super/sub-elements, (ii) in the union (intersection) of the sets of words occurring in the vicinity of its conjunctive (respectively, disjunctive) elements, (iii) in the complement of the intersection of the sets of words occurring in the vicinity of its disjoint elements.

Therefore, we identify the following types of features:

Type 2: Concerning type 2, f_i (as Fig. 5 shows) for a pair of concepts (C^1, C^2) is defined as follows:

$$f_i = \begin{cases} 0, & \text{if } fr_i^1 = 0 \text{ and } fr_i^2 = 0 \\ 1, & \text{if } fr_i^1 \neq 0 \text{ and } fr_i^2 = 0 \\ 2, & \text{if } fr_i^1 = 0 \text{ and } fr_i^2 \neq 0 \\ 3, & \text{if } fr_i^1 \neq 0 \text{ and } fr_i^2 \neq 0 \end{cases}$$

For example, the training pair (Citation-O_1, Proceedings-O_1) in Fig. 1 is represented by the vector [1, 1, 2, 2, 2, 0, ..., 0], where the first five features correspond to the following words: citation, to, proceedings, title, # of pages. All other features correspond to words that do not appear in any of the constituent concepts of the pair and as a result their value is zero.

Type 3: Concerning type 3, f_i (as Fig. 5 shows) for a pair of concepts (C^1, C^2), is defined as follows:

$$f_i = \gamma \times fr_i^1 + (1 - \gamma) \times fr_i^2 \quad (1)$$

where $\gamma \in (0, 0.5)$ is introduced so as to indicate the direction of the subsumption relation. If $\gamma = 0.5$ then symmetric pairs have an identical representation. Since γ is an external parameter, we have studied its effect in the classification task in the experiments conducted.

Type 4: For the 4th type, the TF/IDF [30] values (w_i^j) of the extracted words are being used, instead of their frequencies (fr_i^j). Specifically, f_i is defined as follows:

$$f_i = \gamma \times w_i^1 + (1 - \gamma) \times w_i^2$$

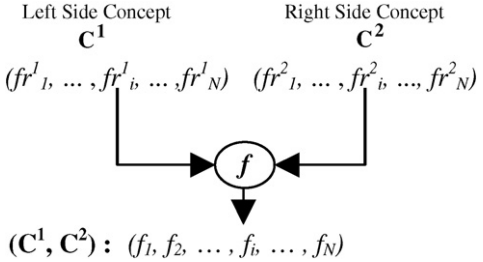

Fig. 5. Function f.

where, γ is as in equation (1), and for a concept C^j, $j = 1,2$ in a pair (C^1, C^2), w_i^j is defined as follows:

$$w_i^j = fr_i^j \times idf_i$$
$$idf_i = \log_2 \frac{\#C}{n_i}$$

fr_i^j is the frequency of the ith word, idf_i is the inverse of the percentage of the concepts that "contain" the ith word in their vicinity, $\#C$ is the total number of concepts in both source and target ontologies and n_i is the number of concepts ($0 < n_i < \#C + 1$) that "contain" the ith word at least one time.

Type 5: In order to overcome the problem of symmetric pairs' identical representation without introducing the extra parameter γ, we can represent concept pairs as feature vectors of size $2 \times N$. Each vector is of the form $(f_1, f_2, \ldots, f_N, f_{N+1}, f_{N+2}, \ldots, f_{2N})$, where f_i, $i = 1, \ldots, 2N$ is defined as follows:

$$f_i = \begin{cases} fr_i^1, & \text{if } 0 < i < N + 1 \\ fr_{i-N}^2, & \text{if } N < i < 2N + 1 \end{cases}$$

In plain words, the feature vector that represents the concept pair is the concatenation of the vectors that represent the constituent concepts of the pair.

Type 6: Similarly with features of type 5, concept pairs are represented as feature vectors of size $2 \times N$. However, f_i (similarly to features of type 4) is defined in terms of the TF/IDF values for the corresponding words:

$$f_i = \begin{cases} w_i^1, & \text{if } 0 < i < N + 1 \\ w_{i-N}^2, & \text{if } N < i < 2N + 1 \end{cases}$$

Beyond properties and words, concepts may be described by means of *latent features*. Specifically, in this case, concepts are transformed to multinomial distributions over latent features. This is done by applying the reverse generative process of the LDA model, as described in Section 3. Concerning any concept of O_1 or O_2, this is represented as a bag of words, extracted from its vicinity. According to the reverse generative process, the resulting multinomial distribution over a given number of T latent features will be of the form $(lt_1, lt_2, \ldots, lt_i, \ldots, lt_T)$, where lt_i, $i = 1, \ldots, T$ corresponds to the ith latent feature and specifies the "contribution" of the corresponding feature in approximating the intended meaning of a concept (i.e. the probability $P(z)$, labeling the arrows in Fig. 3). Therefore, we identify the following types of features:

Type 7: Concerning type 7, the feature f_i (as shown in Fig. 5) for a pair of concepts (C^1, C^2), is defined as follows:

$$f_i = \gamma \times lt_i^1 + (1 - \gamma) \times lt_i^2$$

Type 8: Concerning type 8, similarly with the features of type 5, f_i for a pair of concepts (C^1, C^2), is defined as follows:

$$f_i = \begin{cases} lt_i^1, & \text{if } 0 < i < N + 1 \\ lt_{i-N}^2, & \text{if } N < i < 2N + 1 \end{cases}$$

Studying the importance of properties of concepts 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 equivalence mappings of properties. As far as the use of words 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 computation of equivalence mappings among ontology elements.

Please cite this article in press as: V. Spiliopoulos, et al., On the discovery of subsumption relations for the alignment of ontologies, Web Semantics: Sci. Serv. Agents World Wide Web (2010), doi:10.1016/j.websem.2010.01.001

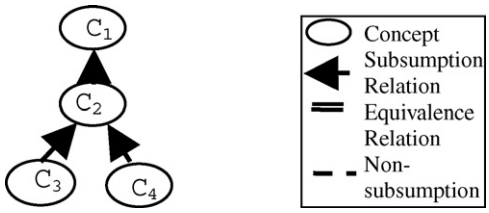

Fig. 6. Simple example.

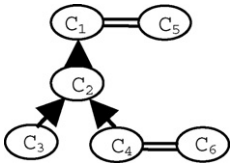

Fig. 7. Stated equivalences.

Finally, the use of latent features (feature types 7 and 8) is investigated to show whether they can capture the intended meaning of concepts more precisely than properties of concepts or words. Similarly with the case of words as classification features, latent features do not require the use of any method for the discovery of equivalence mappings among ontology elements in the source and target ontologies.

4.2. Generating the training dataset

As it has been stated, training examples for classes "$\sqsubset$," and "λ" are being generated by exploiting the source and target ontologies, each one in isolation (i.e. the constituent concepts of each training example pair belong in the same ontology). Training examples can also be generated by exploiting equivalence mappings between the concepts of the two ontologies, in combination with the transitive nature of the subsumption relation. However, as already stated, this is only an option and it is not a necessity for *CSR* (in experiments where equivalence mappings are being exploited, we clearly state it). The paragraphs that follow specify how examples are being generated.[3]

Training Examples for the class "$\sqsubset$". The basic rules for the generation of these examples 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. For example, given the ontologies in Fig. 6, the derived concept pairs are the following: (C_2,C_1), (C_3,C_1), (C_4,C_1), (C_3,C_2), and (C_4,C_2).
- *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. For example, given the ontology in Fig. 7, the resulting pair examples are the ones generated from the previous rule, plus the following pairs: (C_2,C_5), (C_3,C_5), (C_4,C_5), (C_6,C_1), (C_6,C_2), and (C_6,C_5).

[3] In contrast to the rules presented in this section, we have also used a reasoner to generate the training pairs of classes "$\sqsubset$" and "λ". This alternative was not adopted as there was no improvement to the results, while the execution time of *CSR* was unacceptable for large ontologies. Moreover, the different categories of example pairs are exploited for the balancing of the dataset.

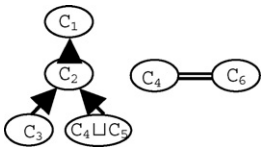

Fig. 8. Disjunction of concepts.

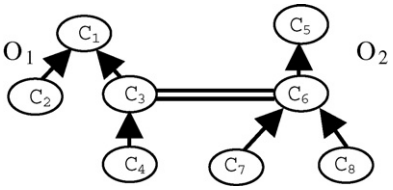

Fig. 9. Concepts of different ontologies.

- *Disjunction of concepts.* Enrich the set of pairs by exploiting the disjunction construct in the definition of concepts: When one concept is specified to be the disjunction of others (e.g. the concept $C_4 \sqcap C_5$ in Fig. 8), and it is specified to be subsumed by another concept (e.g. by the concept C_2 in Fig. 8), then each concept in the disjunction is subsumed by the more general one (i.e. it holds that $C_4 \sqsubset C_2$ and $C_5 \sqsubset C_2$). Therefore, in Fig. 8, the generated example pairs by this rule are (C_4,C_2) and (C_5,C_2). By taking into account also the equivalent concepts rule, the concept C_4 can be substituted by its equivalent concept, and therefore, the pair (C_6,C_2) is included as well.
- *Equivalent concepts of different ontologies* (*i.e. equivalence mappings*). As already stated, we may optionally create training examples for the class "$\sqsubset$," by exploiting equivalence mappings between concepts in different ontologies. In the example depicted in Fig. 9, where concepts C_3 and C_6 are mapped as equivalent, the training examples that can be generated are as follows: (C_4,C_5), (C_4,C_6), (C_3,C_5), (C_7,C_3), (C_7,C_1), (C_8,C_3), (C_8,C_1), and (C_6,C_1).

Training examples for the class "λ". 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 "λ". Four basic rules summarize the generation of examples for the class "λ" and define different categories of training examples:

- *Siblings at the same hierarchy level.* This includes pairs of concepts that are siblings (i.e. 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 "λ". This category can also be enriched by exploiting equivalences of concepts and disjunctions. For example, given the ontology in Fig. 10 the resulting examples are the following: (C_2,C_3), (C_2,C_4), (C_2,C_5), (C_3,C_5), (C_4,C_5), (C_6,C_8), and (C_7,C_8). Symmetric pairs are also included in this category.

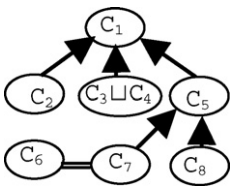

Fig. 10. Siblings.

Please cite this article in press as: V. Spiliopoulos, et al., On the discovery of subsumption relations for the alignment of ontologies, Web Semantics: Sci. Serv. Agents World Wide Web (2010), doi:10.1016/j.websem.2010.01.001

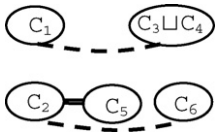

Fig. 11. Object properties.

- *Siblings at different hierarchy levels*. If any concept that is in a pair belonging in the "siblings at 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 to the previous categories, this one also can be enriched by exploiting disjunctions of concepts and equivalence relations between them. For example, given the ontology in Fig. 10 the pairs that are generated are as follows: (C_2,C_6), (C_2,C_7), (C_2,C_8), (C_3,C_6), (C_3,C_7), (C_3,C_8), (C_4,C_6), (C_4,C_7), and (C_4,C_8). Symmetric pairs are also included in this category.
- *Concepts related through non-subsumption relation*. This includes concepts that are related via an object property and are not related with a subsumption relation or concepts related via an equivalence relation. As with the previous categories, this category may also be enriched by considering disjunctions and equivalences between concepts. For example, given the specifications in Fig. 11 the pairs that are generated are as follows: (C_1,C_3), (C_1,C_4), (C_2,C_6), (C_5,C_6), (C_2,C_5) and their symmetric pairs. When equivalence mappings are exploited, this category also includes these mapping pairs.
- *Inverse pairs of concepts that are related with the subsumption relation*. All concept pairs (C_2,C_1) such that C_1 is subsumed by C_2, but C_2 is not subsumed by C_1, constitute examples for the class "λ".

4.3. Creating a balanced dataset

As it is evidenced by the above, the number of training examples for the class "⊏," (minority class) is less than the number of examples for the class "λ" (majority class). As mentioned in Section 3.3, it is very important for the performance of the classifier that the training examples for both classes to be balanced in numbers.

The first strategy adopted to deal with the imbalance problem is the *variation of random under-sampling*. According to this strategy, the training examples are chosen as follows:

1. All examples for the class "⊏," are included.
2. Generate examples for the class "λ":
   2. 1 Examples that are repeated across different categories for the class "λ" are removed.
   2. 2 Select n/t examples from each category of training examples for the class "λ" randomly. n is the number of examples in class "⊏," and t is the number of different categories of class "λ".

The second strategy is the *random over-sampling*. According to this strategy, the training examples are chosen as follows:

1. All examples for the class "λ" are included.
2. Generate examples for the class "⊏,":
   2. 1 Randomly select examples for the class "⊏,", until the two classes contain equal number of examples.

The third strategy exploits the semantics of the input ontologies to generate new *artificial training examples* for the minority class "⊏,", instead of simply performing *random over-sampling* that adds duplicate examples for the class "⊏,". Specifically, this strategy is as follows:

1. All examples for the class "λ" are included.
2. Generate examples for the class "⊏,":
   2. 1 All n examples for the class "⊏," are included.
   2. 2 Generate a artificial examples for the class "⊏,", until $a+n=r$, where r is the number of the examples for the class "λ".
   2. 3 If $a+n<r$, then perform random over-sampling to the original n examples for the class "⊏,", until $a+n=r$.

Artificial examples are generated as follows: Given a set of concepts $C=\{C_1, C_2, \ldots, C_m\}$ with $C_1 \sqsubset F$, $C_2 \sqsubset F$, …, $C_m \sqsubset F$, then the disjunction (⊓) or conjunction (⊔) of all combinations $\mathbf{C}(|C|,i)$ of i concepts in C, generate concepts that are subsumed by the concept F (e.g. $C_1 \sqcap C_2 \sqsubset F$, $C_1 \sqcap C_2 \sqcap C_m \sqsubset F$, or $C_3 \sqcup C_m \sqsubset F$). Therefore, these constitute training examples for the class "⊏".

In our case, the disjunction (⊓) and conjunction (⊔) of concepts is performed on their vector representation. Specifically, similarly to Section 4.1 given two concepts A and B represented as vectors $(fr_1^j, fr_2^j, \ldots, fr_N^j)$, where fr_i^j, $i=1,\ldots,N$, corresponds to the frequency (respectively, TF/IDF value) of the distinct N words extracted from O_1 and O_2, $j=A, B, A \sqcup B$ (respectively $A \sqcap B$) is represented by $(fr_1^C, fr_2^C, \ldots, fr_N^C)$ (respectively $(fr_1^D, fr_2^D, \ldots, fr_N^D)$). fr_i^C and fr_i^D are defined as follows:

$$fr_i^D = \begin{cases} 0, & \text{if } f^A r_i = 0 \text{ and } f^B r_i = 0 \\ 1, & \text{if } f^A r_i \neq 0 \text{ or } f^B r_i \neq 0 \end{cases} \quad (2)$$

$$fr_i^C = \begin{cases} 0, & \text{if } f^A r_i = 0 \text{ or } f^B r_i = 0 \\ 1, & \text{if } f^A r_i \neq 0 \text{ and } f^B r_i \neq 0 \end{cases} \quad (3)$$

It should be pointed out that the strategy for the generation of *artificial training examples* is applied only when the concepts in a pair are represented as vectors of frequencies or TF/IDF values. This is so because: (a) Statistically generated latent features are always non-zero (i.e. $fr_i^j \neq 0$), which means that $f^B r_i$ and $f^A r_i$ will always have non-zero values, resulting always to the same vector for all conjunctions and disjunctions: (1, 1, …, 1). (b) Equations (2) and (3) cannot be applied when properties are used for the generation of classification features (i.e. features of type 1): in this case we have not defined a vector representation for a concept.

4.4. Pruning the space of combinations

Taking into account the semantics of the subsumption relation, it is possible to narrow the search for concept pairs that belong to the subsumption class. In other words, instead of generating all possible concept pairs from both ontologies, we may prune the space of possible concept pairs by excluding pairs of concepts for which a subsumption relation can not 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* might not have subconcepts, hence are called *unit concepts*. We consider that an ontology may include more than one subsumption hierarchies for concepts.

First, to prune the search space, the proposed algorithm checks whether any concept from the source ontology is subsumed by any of the unit concepts of the target ontology, and then by any root concept.

If a pair is not classified in the class "⊏", then the hierarchy rooted by the corresponding concept of the second ontology is not examined by the classifier.

If a pair is assessed to belong to the class "⊏", then recursively, the algorithm tests whether the concept from the source ontology is subsumed by the direct subsumees of the corresponding

Please cite this article in press as: V. Spiliopoulos, et al., On the discovery of subsumption relations for the alignment of ontologies, Web Semantics: Sci. Serv. Agents World Wide Web (2010), doi:10.1016/j.websem.2010.01.001

concept in the target ontology. This happens until either a pair is assessed to belong in the class "λ", or until the leaf concepts are reached.

5. Experimental results and discussion

5.1. The datasets

Given that there are no datasets for the evaluation of methods concerning the subsumption computation problem, we are evaluating *CSR* using three existing datasets for evaluating alignment methods. For these datasets we have extended the gold standard specifying their alignment, by including subsumption relations among their concepts (i.e. subsumption mappings). The compiled datasets are available at the URL http://www.icsd.aegean.gr/incosys/csr.

The first dataset has been derived from the benchmarking series of the OAEI contest [31]. As our method exploits the properties of concepts (for the cases where properties are used as features), we do not include the OAEI ontologies whose concepts have no properties. Furthermore, we have excluded from the dataset the OAEI ontologies with no defined subsumption relations among their concepts (i.e. those in which there is no hierarchy of concepts). This is done because the proposed method exploits the subsumption relation in the source and target ontologies to generate training examples.

More specifically, all benchmarks (101–304) except those in categories *R1–R4*, define the second ontology of each pair as an alteration of the same ontology (i.e. the first one, numbered 101). The benchmarks can be categorized based on their common features (as alternations of 101) as follows: (a) in categories *A1–A5* (210, 237, 238 and 249), the lexicalizations of the elements in the target ontologies have resulted from various changes/replacements (uppercasing, underscore, foreign language, synonyms or random strings) of the corresponding lexicalizations in 101, (b) in categories *A6–A7* (225 and 230) the local restrictions in properties (defined using the <*owl:Restriction*> construct) have been removed, some of the properties have been modeled in more detail, and some unit concepts (i.e. concepts with no subsumees) have been removed, (c) in categories *F1–F2* (222, 237, 251 and 258) the hierarchies have been pruned, resulting in more flat ontologies, and in *F2* random lexicalizations of all elements have been introduced, as well. Finally, (d) in categories *E1–E2* (223, 238, 252 and 259), similarly to *F1–F2*, hierarchies have been expanded in depth, and in *E2* random lexicalizations of all elements have been introduced, as well. Extensive information concerning the benchmark series is provided in the OAEI 2006 contest site [31].

The second dataset is composed of pairs of real-world ontologies concerning course catalogs of the Washington and Cornell Universities. These are available in the *Illinois Semantic Web Archive* [32]. Each university provides two versions of its course catalog: an extended version and a mini version. By utilizing all combinations of the catalogs, the dataset is composed of six pairs of course catalogs. More specifically, courses are organized into schools and colleges, then into departments and centers within each college. Each course is described by a textual description. The reason for selecting these ontologies was that due to their real-world usage, they contain textual descriptions of the courses they classify. Details about the ontologies' statistics are presented in Table 1.

The third dataset is composed of 45 pairs of real-world ontologies coming from the Consensus Workshop track [33] of the OAEI contest 2006 (pairs result from all combinations per two). The domain of the ontologies concerns the organization of conferences and they have been developed within the OntoFarm project [34]. Detailed information concerning these ontologies is provided in the Consensus Workshop track site [33].

The gold standard for all datasets has been manually created by a knowledge engineer. The major guidelines that were followed by the engineer in order to find the subsumption relations are as follows: (a) use existing equivalence mappings in order to find inferred subsumptions, and (b) understand the "intended meaning" of the concepts (e.g. by inspecting specifications and relevant information attached to them), and use common sense to locate any subsumption relations that cannot be inferred by the existing equivalences. The format of the gold standard is the same with the one used in the benchmark series of the OAEI competition (more information is provided at http://people.kmi.open.ac.uk/marta/oaei09/orientedMatching.html).

5.2. Experiments and results

Results show the *F-measure*, *Precision* and *Recall* of the proposed method as it is applied in the ontology pairs specified in Section 5.1. *F-measure* is the ratio $2 \times Precision \times Recall/(Precision+Recall)$, where *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 each pair of ontologies, using each of the classifiers: C4.5, knn, NaiveBayes (Nb) and Svm. For each of the classifies we have run 51 experiments using the alternative feature types defined in Section 4.1, in combination with a dataset balancing method (over-sampling, under-sampling, and synthetic pairs, denoted by "*ov*", "*un*", and "*syn*" respectively), and for different values for the parameter γ (for features of type 3, 4 and 7) ranging in $\{0.1, 0.2, 0.3, 0.4\}$. Subsequently, we denote the *CSR* configuration used in an experiment by CT+FT+DB+γ, where CT is the classifier, FT is the feature type number, DB is the type of the dataset balancing method used, and γ is the value of the parameter γ for the feature types that use this parameter.

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 "$\sqsubset$,", 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 *CSR* and the baseline classifier exploit the same information. Two different types of experiments are presented concerning the baseline classifier: The one using features of type 1 (denoted by Baseline+Props) and the other features of type 2 (denoted by Baseline+Words). The other experimentation settings (exploiting other feature types) achieved too low results (<8% in terms of average *F-measure*), something which was rather expected.

To investigate whether, given a set of equivalence mappings, a reasoning mechanism suffices for the purpose of computing subsumption mappings among the elements of distinct ontologies we also compared the effectiveness of *CSR* with that of a Description Logic reasoning engine.[4] In order for the reasoner to be able to infer subsumption mappings between concepts of the source and target ontologies we specify axioms concerning only equivalence mappings of properties (Reasoner+Props), or alternatively, equivalences mappings of both properties and concepts (Reasoner+Props+Con).

Here we have to state that although there are ontology alignment methods that are able to locate subsumption mappings

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

Please cite this article in press as: V. Spiliopoulos, et al., On the discovery of subsumption relations for the alignment of ontologies, Web Semantics: Sci. Serv. Agents World Wide Web (2010), doi:10.1016/j.websem.2010.01.001

Table 1
Statistics for the second dataset.

Course catalog	#Concepts	Depth	#Instances	#Max. siblings of concept	#Min. siblings of concept	#max instances of concept	#min instances of concept
Washington	166	4	6950	49	2	212	5
Washington Mini	39	4	1912	11	2	214	2
Cornell	170	4	4360	27	1	161	5
Cornell Mini	33	4	1526	10	2	155	2

(already presented in Section 2.2), their main target is the computation of equivalent mappings. For this reason, the evaluation of these methods focuses on their efficacy on computating equivalence mappings. Moreover, in contrast to *CSR* which computes strict subsumption relations, these methods compute subsumption relations in general: A fact that makes the results of *CSR* not directly comparable to the ones produced by these methods.

Furthermore, as far as we know, until recently, there were no corpora (datasets) for evaluating the computation of subsumption mappings. This is true, despite the known usefulness of subsumption (and generally, ordered) relations in the ontology alignment community. The first published evaluation concerning subsumption mappings is presented in Ontology Alignment Evaluation Initiative 2009,[5] in a specialized track named Oriented Matching.[6] The corpora used for the evaluation of the participating methods are the ones we created in the course of this work (and thus, being biased from our point of view[7]) and have been used in the evaluation of *CSR*, which is presented in the next Sections. The systems that participated in this track gave results only for the first dataset, derived from the benchmarking series of the OAEI contest, as described in the previous paragraph. More details concerning the results of the track are available in Section 5.6.

Concerning the implementation and the parameters of the machine learning classifiers we used in the conducted experiments, the following apply: (a) We employ the Naïve Bayes classifier from the Weka toolkit, as implemented in [21]. (b) We have used the IBk class from the Weka toolkit, which implements a knn classifier as presented in [22] with value of k equal to 2. (c) Concerning the Svm classifier we have used the libSVM [23] implementation with its default values and radial basis function as kernel. (d) Finally, we employ the j48 [24] implementation of the C4.5 [35] decision tree learning algorithm, configured with Weka's default values.

At this point we must recall that when *CSR* exploits words for computing the features of concept pairs (i.e. in all feature types, except type 1), then no equivalence mappings are required. However, if any experiment has been performed by exploiting computed equivalence mappings among the elements of the input ontologies, this is clearly stated: Specifically, this is indeed the case only in the third and more challenging dataset (Section 5.5).

5.3. Results in OAEI benchmark series

In this section we present the results of the proposed method, applied to the OAEI benchmark series dataset. We provide a comparative analysis of the different configurations of *CSR*, for different types of classifiers, and we focus on the results of the configuration with the best performing classifier. Furthermore, we present the results of the SEMA mapping tool (for computing equivalence mappings between ontology elements) so as to thoroughly discuss the results of the reasoner, which exploits equivalence mappings between properties and concepts. Concerning the representation of pairs of concepts, in this dataset equivalence mapping of properties are exploited for the generation of classification features only in the case of feature type 1 (classification features are based on properties of concepts), as explained in Section 4.1. However, in this dataset no equivalence mappings are exploited by *CSR* for the generation of training examples of class "$\sqsubset$,".

Fig. 12 depicts the *F-measure* achieved by the "best" configurations of *CSR*, for each one of the classifiers, and for all different categories of the dataset. The average *F-measures* over all the categories are depicted in the right-most part of the diagram. We observe that the configuration with C4.5 outperforms, or performs equally well to, all other configurations, in all test categories and on average, for each dataset category.

This behavior can be explained by the specific inherent features of decision tree classifiers [24]: (i) disjunctive descriptions of cases fits naturally to the subsumption computation problem. This is true since more than one feature may indicate whether a specific concept pair belongs in the class "$\sqsubset$,". (ii) Decision trees are tolerant to errors in the training set. 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 equivalence mappings among properties is imprecise, or more generally, the representation of the example pairs (i.e. the feature vectors) may not be accurate enough. For example, the words extracted from the constituent concepts of a pair may not be enough, or they may not be representative for this pair, for the subsumption computation problem.

Finally, as already stated, all classifiers are used with their default parameters and as proven by the results, C4.5 adapts successfully into the dataset without the need of modifications in its settings: A fact very important for real-world application scenarios, where the need for tuning the methods to specific pairs of ontologies or domains would be a major obstacle to their successful application.

Fig. 13 depicts how each best performing configuration is influenced (in terms of the *F-measure* achieved) by the dataset sampling method used ("?" indicates each of the sampling methods shown in columns). As explained in Section 4.3, when properties of concepts are exploited as classification features, the synthetic sampling method cannot be applied. According to the literature [25,26], there is no perfect sampling method and the effectiveness of these methods varies for different classifiers and different datasets. Moreover, as it is also shown, random over- or under-sampling achieves competitive or better results than more sophisticated methods. As Fig. 13 shows, this is also the case for *CSR*: We observe that C4.5 is heavily influenced by the sampling method, with oversampling achieving by far the best performance. On the other hand, we observe that knn and Svm classifiers perform better with the synthetic sampling method.

The best results are achieved by *CSR* with the C4.5 classifier. As a result, we focus on configurations with this classifier, for a more in depth analysis of the *CSR* method.

In Fig. 14 we present the number of pairs classified by each one of the three best performing configurations, which in contrast to what *CSR* assesses, belong to the equivalence rather than to the

[5] http://oaei.ontologymatching.org/2009/.
[6] http://people.kmi.open.ac.uk/marta/oaei09/orientedMatching_Results.html.
[7] We constructed this set of corpora assuming that no equivalence mappings between concepts shall be exploited: Subsumptions can be easily inferred by exploiting equivalence mappings.

Please cite this article in press as: V. Spiliopoulos, et al., On the discovery of subsumption relations for the alignment of ontologies, Web Semantics: Sci. Serv. Agents World Wide Web (2010), doi:10.1016/j.websem.2010.01.001

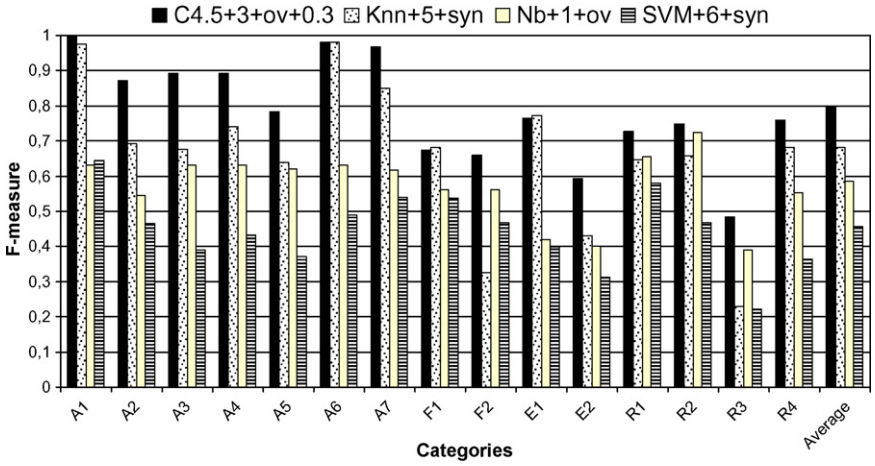

Fig. 12. Best experiment of each classifier.

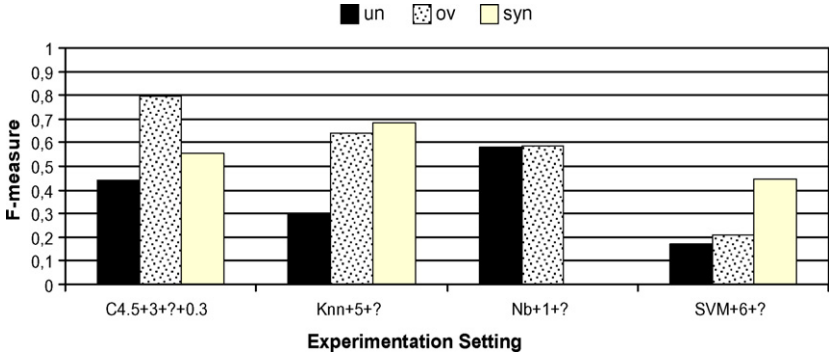

Fig. 13. Sensitivity of classifiers to the sampling method.

subsumption relation (i.e. the misclassified cases). As it is shown, the best performing configuration (C4.5, exploiting words' frequencies, with over-sampling and $\gamma = 0.3$), has less than one errors in all categories. This is a really important feature of *CSR* as it can be used for "filtering" the results of systems that compute equivalence mappings effectively: This is important, since, as it has already been stated, state of the art systems tend to confuse subsumption relations with equivalence ones [2].

Fig. 15 and Fig. 16 present the *Precision* and *Recall* of the best performing configurations of *CSR*, in all test categories, in comparison with the best performing configuration of the baseline classifier and the two configurations of the DL reasoner-based classifier.

A first observation is that the best performing *CSR* configuration (C4.5+3+ov+0.3) achieves for all the categories, on average, the best precision, compared to all the other experimentation settings. In terms of recall, the three *CSR* configurations perform almost equally well, but the Reasoner+props+con configuration outperforms them. Here we have to point out that the three *CSR* configurations do not exploit equivalence mappings among concepts or properties, while the Reasoner+props+con exploits equivalence mappings for both, properties and concepts. Last but not least, we have to point out that there are concept pairs for which the subsumption relation holds among them and only *CSR* manages to locate them. In category A2, for example, we observe that *CSR* configurations have a higher recall than the two reasoner-based classifiers, and still perform better or equally good in terms of precision.

Furthermore, as Fig. 15 and Fig. 16 depict, the experiments contacted with the different configurations of the *CSR* method achieve on average a better recall and a better precision than the best performing base line classifier (Baseline+Props). This means that

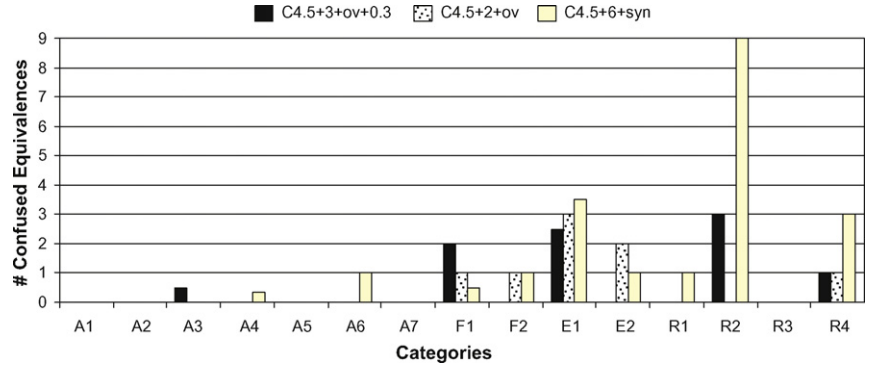

Fig. 14. Confused Equivalences of *CSR* using C4.5.

Please cite this article in press as: V. Spiliopoulos, et al., On the discovery of subsumption relations for the alignment of ontologies, Web Semantics: Sci. Serv. Agents World Wide Web (2010), doi:10.1016/j.websem.2010.01.001

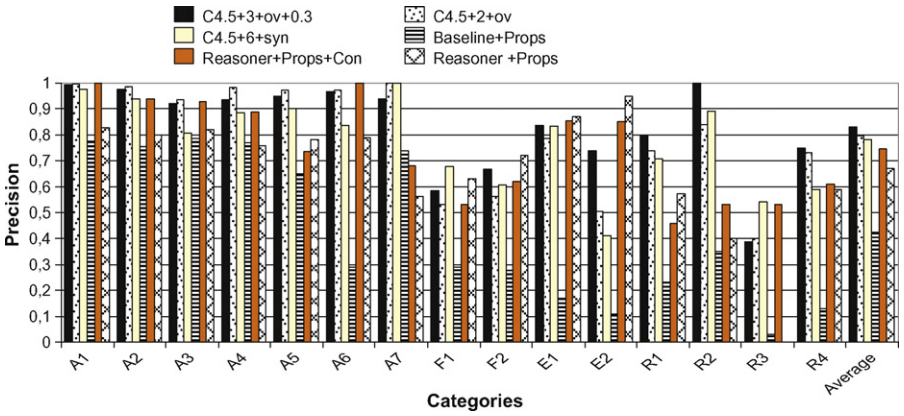

Fig. 15. Precision for the different categories.

classifiers do generalize, as they manage to locate subsumptions that are not in their training dataset.

Another important point is that the three best performing *CSR* configurations (Fig. 15 and Fig. 16) use features generated from the words extracted from concepts (feature types 2, 3, and 6). Therefore, although they do not require equivalence mappings among ontology elements, their performance is influenced by the words extracted. Generally, the pattern in the experiments contacted is that as more words are extracted, better results are achieved by the classification task. This happens since more words lead to more "rich"/"representative" feature vectors of pairs of concepts. Let us for instance consider the category A5 in Fig. 16, where labels of concepts are replaced by random strings and in some cases also comments and instances have been removed from concepts: we observe that the recall (66%) is much less than this in categories (A1–A4) where more words are available. This is further evidenced by the experiments performed in the second dataset, where the ontologies have many defined instances and as a result the extracted words are abundant. We further comment on this in the next subsection.

On the other hand, the Reasoner+Props+Con classifier performs well, provided that it exploits correct equivalence mappings and the structures of the two input ontologies are similar or the same (similar conceptualizations). However, in the real world cases (R1–R3), we observe low precision values (in Fig. 15: 51%, 52% and 61%, respectively). In these cases, the input ontologies model the domain quite differently (concerning the hierarchy of concepts), and also the precision of SEMA is quite low (shown in Fig. 17).

Another interesting case is category A7, where some unit concepts have been removed from the target ontology and the remaining concepts have more properties. Actually, the properties in many ontologies of this category have been replaced by more detailed specifications (e.g. the property "date" has been replaced by the properties "day", "month" and "year"). In this case we observe that SEMA scores 78% in *Precision* and 100% in *Recall* (concerning both concepts and properties in Fig. 17, and only properties in Fig. 18), but Reasoner+Props+Con (respectively Reasoner+Props) achieves 69% (respectively 58%) *Precision* and 100% (respectively 66%) *Recall*. This shows that even when the hierarchies of the aligned ontologies are almost the same, a few erroneously assessed equivalence mappings can deteriorate the performance of the reasoner. Moreover, in category A7 the best performing *CSR* configuration (Fig. 15 and Fig. 16) achieves 97% *Precision* and 100% *Recall*.

The above stated conclusions are further evidenced by the ROC analysis [36] of our experimental results, shown in Fig. 19. In our case, ROC analysis indicates the effectiveness of the best performing *CSR* configuration in classifying testing examples in the distinct classes "$\sqsubset$," and "λ". 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. In our case, the average values are above 0.8, indicating that on average the classifiers can be characterized as "good", in terms of the generalization they achieve.

In the following paragraphs we present results from experiments of all *CSR* configurations with the C4.5 classifier (Fig. 20), focusing on the three sampling methods: under-sampling, over-sampling, and synthetic.

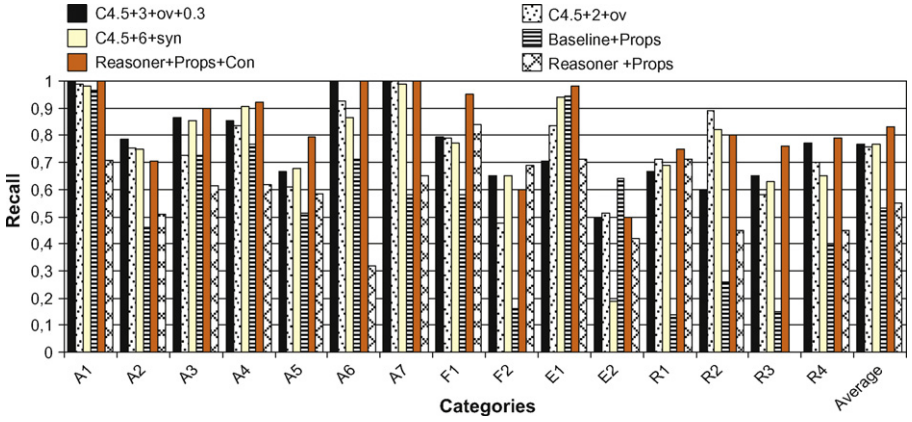

Fig. 16. Recall for the different categories.

Please cite this article in press as: V. Spiliopoulos, et al., On the discovery of subsumption relations for the alignment of ontologies, Web Semantics: Sci. Serv. Agents World Wide Web (2010), doi:10.1016/j.websem.2010.01.001

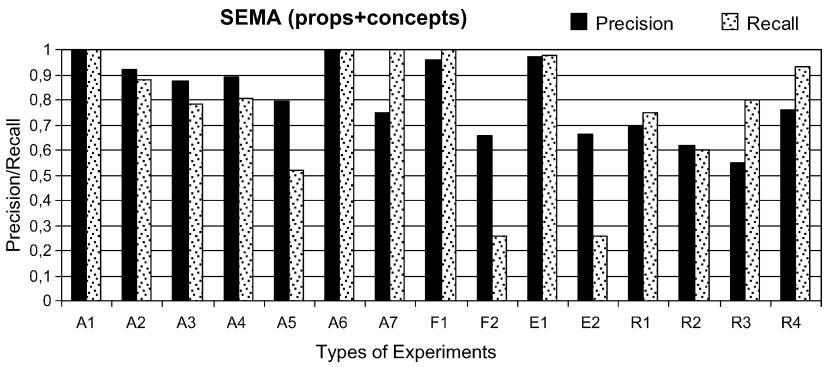

Fig. 17. Precision and recall of SEMA (concepts and properties).

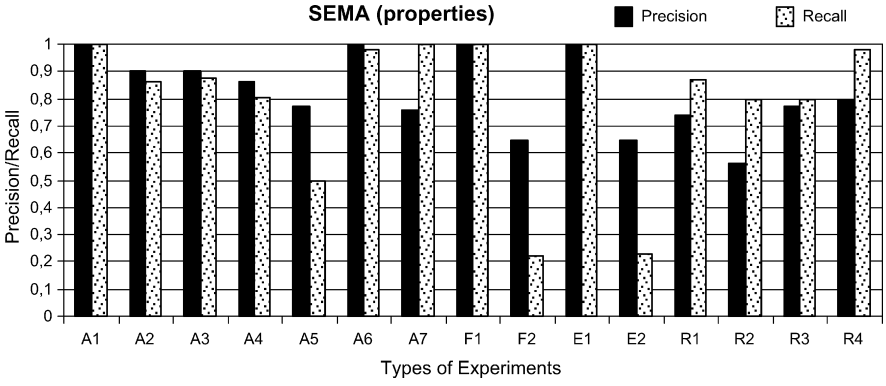

Fig. 18. Precision and recall of SEMA (properties only).

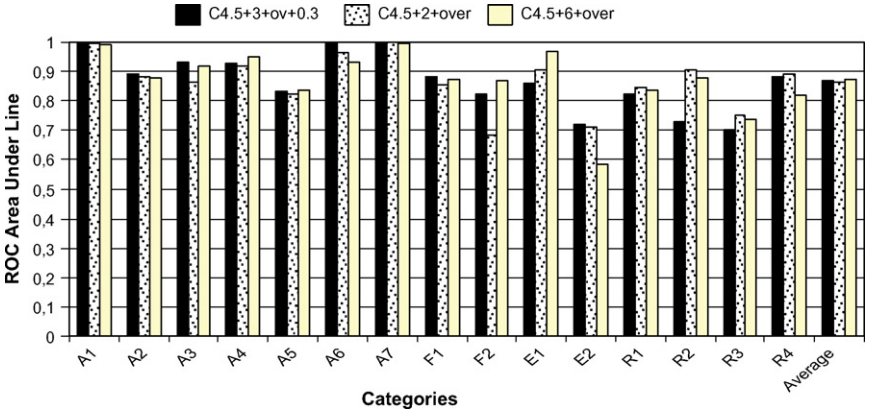

Fig. 19. ROC areas under line in all categories, of best performing.

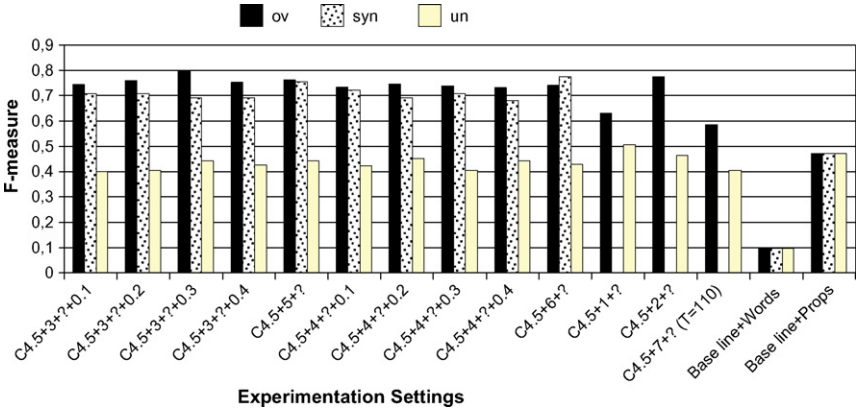

Fig. 20. C4.5 experiments.

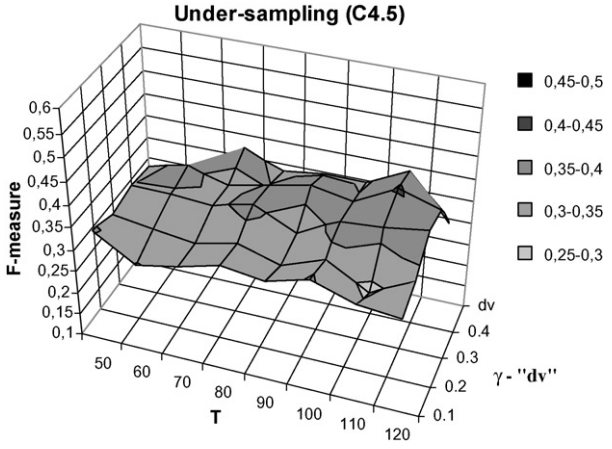

Fig. 21. Latent features analysis—under-sampling.

In Fig. 20 we observe that in contrast to the cases where latent feature are used (presented later in Fig. 21 and Fig. 22), the parameter γ or the usage of double vectors, does not influence the performance of the method (in terms of the *F-measure* achieved) for the same type of classification features. Similarly, we observe that for the same balancing method, the type of features used does not have a major impact on the *F-measure* achieved by the corresponding *CSR* configuration. What seems to have a major impact on the *F-measure* is the dataset balancing method. Specifically, as already shown above, C4.5 is more effective when it generates test cases using over-sampling, rather than synthetic, or under-sampling techniques.

Fig. 21 and Fig. 22 depict the performance of *CSR* when it exploits latent features (features of type 7 and 8). Specifically, Fig. 21 and Fig. 22 depict how the *F-measure* achieved is influenced by the parameters involved, namely: (a) the number of topics (T), (b) the value of the parameter γ for the combination of features of concepts, and (c) the use of double-size vectors (shown as "dv").

The major observation is that again, over-sampling outperforms under-sampling, a fact that is true for C4.5 for any type of features we have experimented with. As already stated in Section 3, C4.5 (and generally, decision tree-based learning methods) tends to perform well with over-sampling techniques, a behavior that it is also evidenced by our experiments for the subsumption computation problem.

In the cases where *CSR* uses feature vectors constructed by means of the computed latent features, the method performs less effectively than in cases where features represent word frequencies and TF/IDF values. This behavior can be explained as follows: (a) Gibbs sampling computes an *approximation* of the latent features (presented in Section 3) that best fit the pseudo-documents corresponding to concepts. (b) The number of topics T affects the approximation achieved. We have experimented with various numbers of topics: Fig. 21 and Fig. 22 present a representative fragment of the experimental results, depicting the best achieved performance. (c) The small number of words appearing in the documents lead the Gibb sampling method to be effective for a small number of topics. However, a small number of topics generate a small number of features, which may not be discriminative enough for the classification task. This is also evidenced by the fact that in the case where double vectors have been used ("dv"), where the number of features in vectors is twice the number of latent features (as explained in Section 4.1), the method performs well in the majority of cases (Fig. 21 and Fig. 22). Finally, (d) the approximation of Gibbs sampling is negatively influenced by the small numbers of words that are available. As we will present in the next dataset where documents contain many words, *CSR* performs quite well when it exploits latent features.

Another interesting observation is that the *CSR* performance in this case (Fig. 21 and Fig. 22) is heavily influenced by the value of parameter γ. More specifically, we observe that when γ equals 0.1 *CSR* tends to perform less efficiently as in the cases where γ is set to higher values. On the other hand, the value of T (number of latent features) slightly influence the performance of *CSR* (i.e. for the same γ, the color and incline of the surface is not heavily altered).

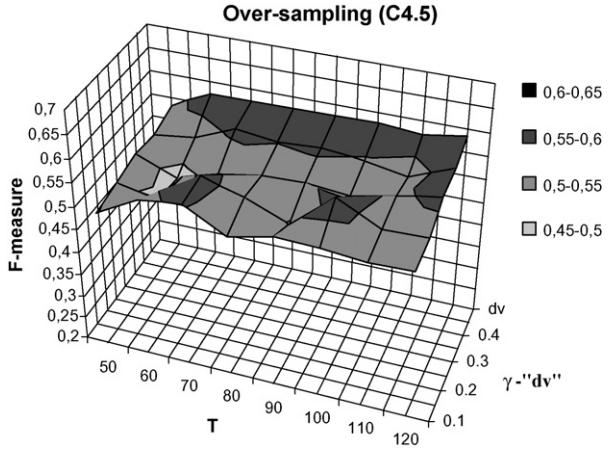

Fig. 22. Latent features analysis—over-sampling.

5.4. Results in the course catalogs of universities

In this section we present the results of *CSR* when it applies to the Course Catalogs dataset. Specifically, we present a comparative analysis of the performance of different *CSR* configurations.

Fig. 23 depicts the performance of the best performing experiments, in terms of *F-measure*, for each one of the classifiers. As in the previous dataset, we observe that configurations with the C4.5 classifier achieve the best results, on average. Furthermore, configurations with the C4.5 classifier outperforms configurations with the other classifiers in all catalog pairs, except in the ontologies (`cornell`, `washington`) where C4.5 is less effective than the Naïve Bayes (78–75%). The quite good performance (in terms of *F-measure*) of C4.5 classifier can be explained using the same arguments as in the Benchmark Series dataset, in addition to the fact that the ontologies belonging to the Course Catalog dataset have many defined instances (see Table 1). For this reason, the number of available words exploited for the generation of feature vectors by the *CSR* method is high, leading to more rich and representative feature vectors for the training example and testing pairs of concepts.

On the other hand, we observe that, on average, the worst performance is achieved by the Naïve Bayes classifier. This can be explained by the fact that the features' independence assumption does not hold for the subsumption computation problem, leading to poor estimation of posteriori probabilities.

Another interesting observation is that when a mini version of a catalog is aligned to the full version of a different catalog (i.e. ontology pairs (`mini_cornell`, `washington`) and (`mini_washington`, `cornell`)), C4.5 achieves more than 81% of *F-measure*. This is a quite important fact, as in these cases ontologies model domain aspects at different granularity levels and there are a lot of subsumption relations between the elements of ontologies.

To further analyze the effectiveness of the classifiers, we present the results of the ROC Area analysis in Fig. 24. In all experiments presented (see Fig. 24), the classifiers perform above 80%, and some

Please cite this article in press as: V. Spiliopoulos, et al., On the discovery of subsumption relations for the alignment of ontologies, Web Semantics: Sci. Serv. Agents World Wide Web (2010), doi:10.1016/j.websem.2010.01.001

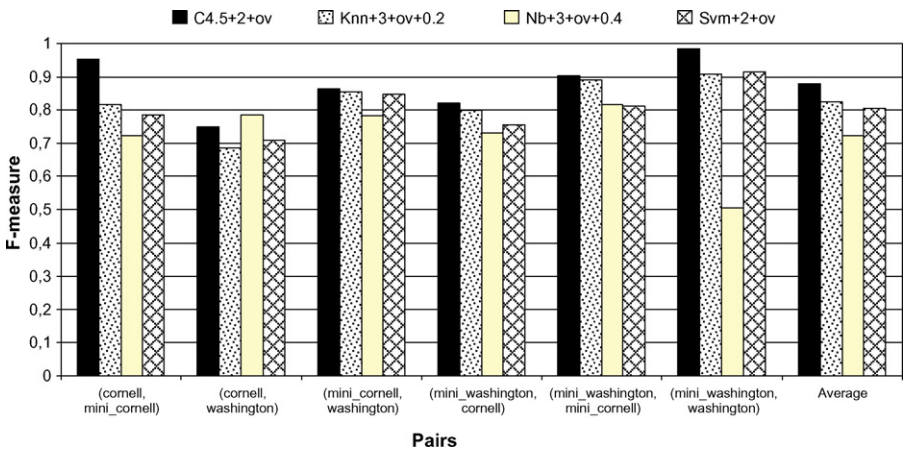

Fig. 23. Best experimentation setting of each classifier.

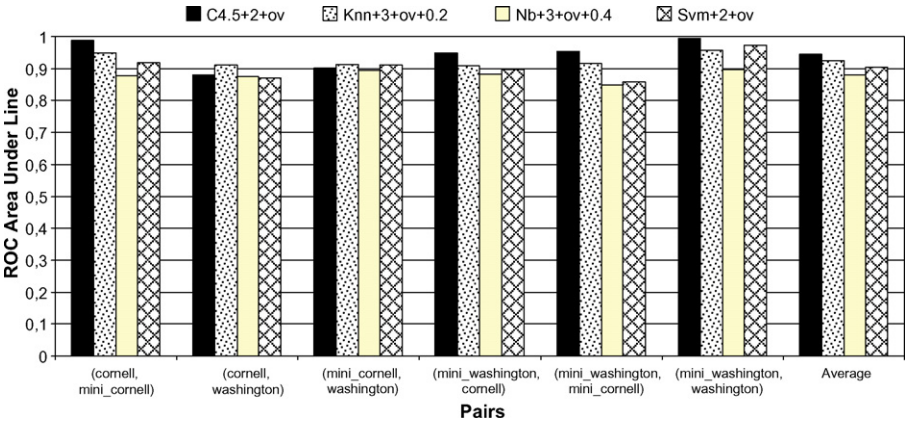

Fig. 24. ROC areas under line of best performing classifiers.

above 90%: A fact that is very encouraging for the effectiveness of *CSR*.

Fig. 25 depicts the *F-measure* achieved by the *CSR* configurations using the latent features computed by the LDA method: C4.5+7+ov ($\gamma \in \{0.1, 0.2, 0.3, 0.4\}$ in Fig. 25) and C4.5+8+ov (double size vectors – indicated as "dv" in Fig. 25 – are used instead of parameter γ). As

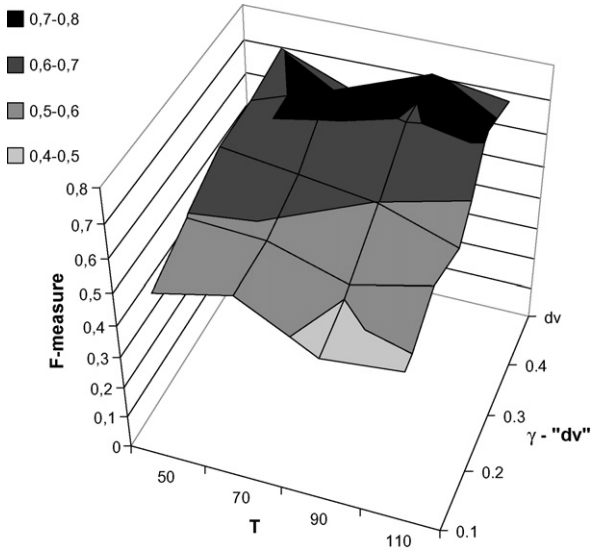

Fig. 25. Latent feature analysis of best performing experiment.

already observed in the benchmark series dataset, we also observe that the performance of *CSR* is heavily influenced by the variations in the value of γ. More specifically, we observe that when γ equals 0.1 *CSR* is less effective than in cases where γ is set to higher values. On the other hand, the value of T (number of latent features used) slightly affects the achieved *F*-measure (i.e. for the same γ, the color and incline of the surface is not heavily altered). The best results are achieved for $\gamma = 0.4$ and $T = 70$ (*F-measure* = 76%). Similar behavior is observed also in the majority of the experiments conducted concerning latent features.

As stated in Section 2.2 oPLMap method locates "similarity" mappings between web directories. These mappings may be interpreted as subsumption mappings. Although *CSR* locates strict subsumption mappings and the results are not directly comparable, we can provide a quantitative comparison between the two methods, as oPLMap is evaluated using the course catalog pair (`cornell`, `washington`). Specifically, the best configuration of oPLMap achieves 68.09% in terms of *F-measure* compared to 77% achieved by *CSR* (Fig. 23) for this particular pair.

5.5. Results in the consensus workshop ontologies

In this section we present the results of *CSR* when it is applied to the Consensus Workshop ontologies. This is a more challenging dataset: The data set includes real-world ontologies with different conceptualizations of their domain, while the words available for the generation of features vectors by the *CSR* method are few (e.g. ontology elements do not have comments or instances).

Please cite this article in press as: V. Spiliopoulos, et al., On the discovery of subsumption relations for the alignment of ontologies, Web Semantics: Sci. Serv. Agents World Wide Web (2010), doi:10.1016/j.websem.2010.01.001

Similarly to the previous datasets, *CSR* with the C4.5 classifier achieves the best results compared to configurations with other classifiers. Therefore, we focus on configurations with this classifier to provide a more in-depth presentation of the results.

Fig. 26 and Fig. 27 depict the *Precision* and *Recall* achieved by the best performing *CSR* configuration: C4.5+5+ov (classifier C4.5 exploiting frequencies of words, in conjunction with the usage of double vectors, using the over-sampling dataset balancing method and also exploiting SEMA's equivalences for the generation of extra training examples). As already stated in Section 4, *CSR* may optionally create extra training examples for the class "$\sqsubset$", by exploiting equivalence mappings between concepts in different ontologies (category "Equivalent concepts of different ontologies"). This is done because of the challenging nature of the dataset.

Firstly, we observe that there are pairs of ontologies in which this configuration of *CSR* performs well in terms of both measures, *Precision* and *Recall*. For example, such pairs are: (`Cmt`, `Iasted`) (*Precision*: 58%, *Recall*: 69%), (`ConfToot`, `Cmt`) (*Precision*: 78%, *Recall*: 81%), (`SoftSem`, `Pcs`) (*Precision*: 64%, *Recall*: 77%), (`Pcs`, `OpenConf`) (*Precision*: 54%, *Recall*: 77%) and (`SoftSem`, `Confious`) (*Precision*: 52%, *Recall*: 67%). On the other hand, in the majority of the ontology pairs, *CSR* performs in a less effective way, leading to an average *Precision* 29.5% and to an average *Recall* 42%. This performance is due to the following phenomena: (a) the lack of words for the generation of feature vectors, in conjunction to the different conceptualizations specified in the source and target ontologies. The latter leads to the need of very representative feature vectors from the classifier to perform efficiently, but the former prohibits such a behaviour, as already discussed in other experiments. (b) SEMA may return erroneous equivalence mappings, leading to wrong training examples for the classifier, which prevent the classifier from properly generalizing. Here we must recall that C4.5 is more tolerant to mistakes than other classifiers: this is evidenced here by its performance in comparison to configurations with other classifiers. However, as it will be explained in the next paragraph, *CSR* performs better when it exploits SEMA's equivalence mappings (both correct and erroneous ones), than in cases where it does not use them.

Furthermore, we must point out that the exploitation of equivalence mappings between the elements of the input ontologies, in combination with the transitive nature of the subsumption relation for the generation of training examples for the class "$\sqsubset$," leads in some cases to increased *Recall*, however sacrificing precision, given also that the equivalences computed may contain many false-positives (e.g. in the (`Cmt`, `Iasted`) pair of ontologies). We do not provide the *Precision* and *Recall* of SEMA as there is no gold standard for the equivalence mappings for this dataset. We must also point out that *CSR* exploits the located equivalence mappings for generating training examples. However, wrong training examples (due to the exploitation of a wrong equivalence mapping between ontology elements), do not necessarily drive *CSR* to infer false-positive subsumption mappings (misclassify concept pairs in the class "$\sqsubset$"). For this reason, we observe that when equivalences are exploited for the generation of training examples, the average *Precision* and *Recall* increase. This is observed in the majority of ontology pairs (in Figs. 26–29).

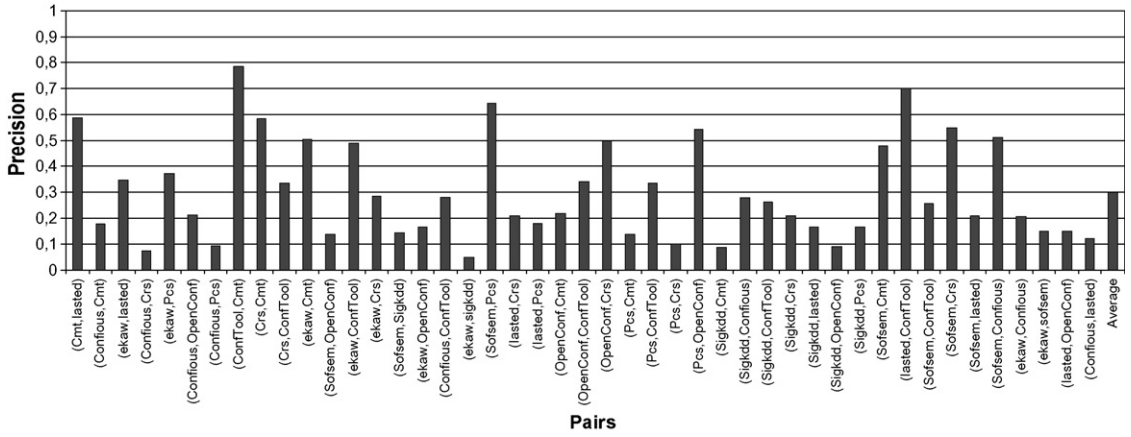

Fig. 26. Precision of *CSR* experiment C4.5+5+ov exploiting equivalences, in all pairs of ontologies.

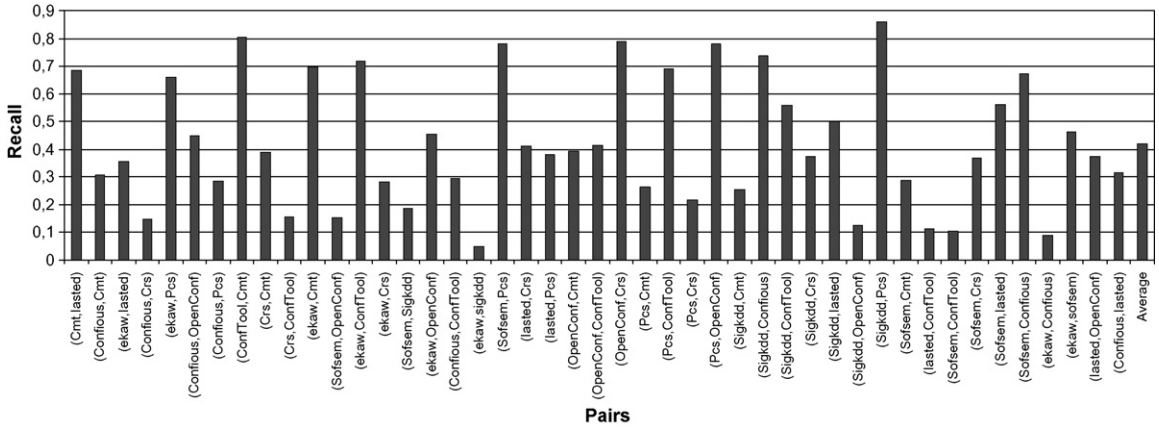

Fig. 27. Recall of *CSR* experiment C4.5+5+ov exploiting equivalences, in all pairs of ontologies.

Please cite this article in press as: V. Spiliopoulos, et al., On the discovery of subsumption relations for the alignment of ontologies, Web Semantics: Sci. Serv. Agents World Wide Web (2010), doi:10.1016/j.websem.2010.01.001

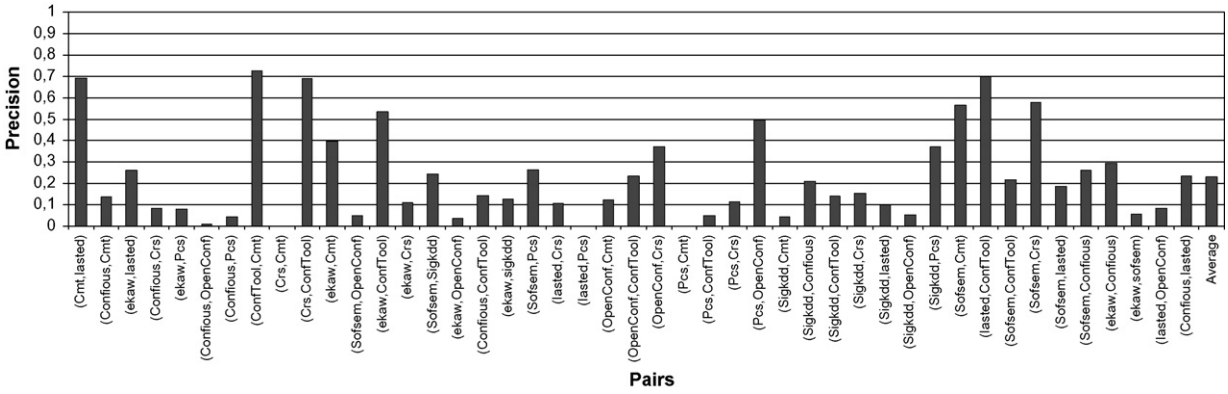

Fig. 28. Precision of *CSR* experiment C4.5+5+ov without exploiting equivalences, in all pairs of ontologies.

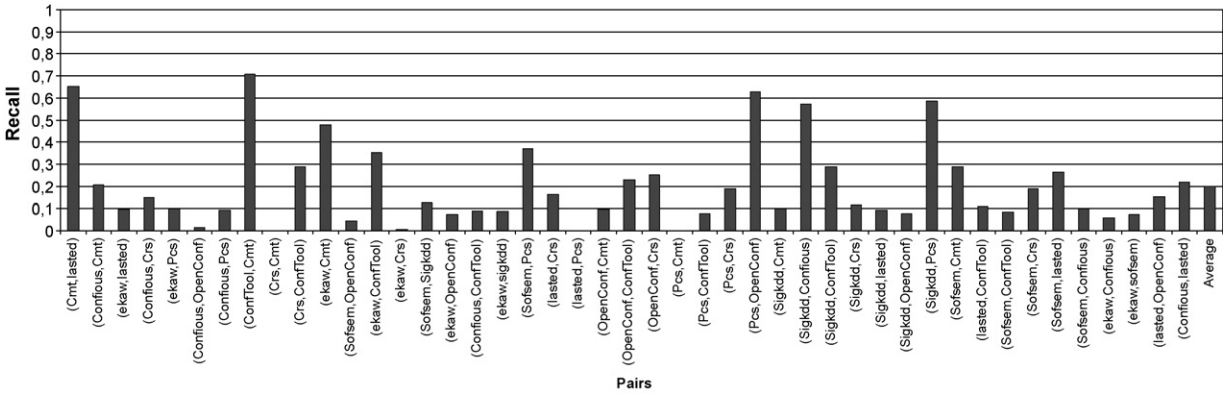

Fig. 29. Recall of *CSR* experiment C4.5+5+ov without exploiting equivalences, in all pairs of ontologies.

Indeed, when no equivalence mappings are exploited for the generation of extra training examples, *CSR* is less effective in terms of *Precision* (Fig. 28) and *Recall* (Fig. 29) in the majority of the ontology pairs, and as a result in the average values (last column). But still there are cases where *CSR* performs quite well in terms of *Precision* and *Recall*: (`Cmt`, `iasted`) (*Precision*: 69%, *Recall*: 65%), (`ConfTool`, `Cmt`) (*Precision*: 73%, *Recall*: 71%) and (`Pcs`, `OpenConf`) (*Precision*: 49%, *Recall*: 63%).

The reason that *CSR* performs well in the above cases (the same applies when equivalence mappings are exploited) is linked to the fact that there are quite many words available in the vicinity of concepts defined in these ontologies. Specifically, factors that influence the amount of words extracted in these cases are: (a) Defined comments in concepts and properties, (b) number of defined properties, (c) number of sub/super-concepts (words for a concept also include words of its sub/super-concepts), or (d) very representative labels that are concatenations of many words (e.g. `Camera_ready_manuscript_deadline`). These are split to words, using the character "_" as a delimiter.

5.6. Results in the Oriented Matching track at the OAEI 2009

The Oriented Matching track comprised two datasets: The first dataset (dataset 1) has been derived from the benchmark series of the OAEI 2006 campaign. As a configuration of *CSR* exploits the properties of concepts (for the cases where properties are used as features), we do not include the OAEI 2006 ontologies whose concepts have no properties. Furthermore, we have excluded from the dataset the OAEI ontologies with no defined subsumption relations among their concepts. This is done because CSR exploits the subsumption relations in the input ontologies to generate training examples. The second dataset (dataset 2) is composed of 45 pairs of real-world ontologies coming from the Consensus Workshop track of the OAEI 2006 campaign (all pairwaise combinations).

The participating methods in the Oriented Matching track that took place in the Ontology Alignment Evaluation Initiative 2009 were: *CSR*, ASMOV, RiMoM and TaxoMap. These systems gave results for the first data set only. We observe that in terms of *F-measure* ASMOV achieves the best results (0.93%), followed by CSR (80%), RiMoM (71%) and then by TaxoMap (0.23%). More details on results are available at http://people.kmi.open.ac.uk/marta/oaei09/orientedMatching_Results.html.

Participating systems exploit equivalence mappings or similarities among elements in order to locate subsumption ones (since they are using the methods for the benchmark track). In contrast to that, *CSR* does not exploit equivalence mappings or direct similarities, computing subsumption relations, directly. This is something very important as explained in Section 1.

6. Conclusions and future work

In this paper we propose the "Classification-Based Learning of Subsumption Relations" method for the alignment of ontologies. *CSR* aims to the computation of subsumption mappings between concepts of two distinct ontologies. This is achieved by alternatively exploiting a variety of different classification features, based on properties or words extracted from the vicinity of concepts in the source and target ontologies. Specifically, *CSR* assesses whether concept pairs of the source and target ontologies belong to the subsumption relation by means of a classification task using state of the art supervised machine learning methods. Given a pair of concepts from two ontologies, the objective of *CSR* is to identify patterns of

Please cite this article in press as: V. Spiliopoulos, et al., On the discovery of subsumption relations for the alignment of ontologies, Web Semantics: Sci. Serv. Agents World Wide Web (2010), doi:10.1016/j.websem.2010.01.001

classification features that provide evidence for the subsumption relation among these concepts. For the training of the classifiers, the proposed method generates examples from the source and target ontologies specifications, tackling also the problem of imbalanced training datasets.

The conclusions that have been drawn by the experimental results can be summarized as follows: (a) *CSR* generalizes effectively over the training examples, locating subsumption mappings that cannot be located by a reasoning mechanism. In these cases *CSR* does not exploit equivalence mappings among the elements of the input ontologies. (b) *CSR* effectively discriminates among subsumption and equivalence mappings. This is important as *CSR* can be used to filter the results of any tool that locates equivalence mappings. (c) The most deciding aspects for the performance of *CSR* are the classifier and the dataset balancing method. Specifically, C4.5 outperforms all other tested classifiers in all datasets. Concerning the balancing method, over-sampling outperforms under-sampling and synthetic sampling when it is applied in combination with C4.5. In this case *CSR* achieves the best performance: A fact quite convenient due to the computational complexity of the synthetic sampling. (d) The synthetic sampling outperforms the other sampling methods when it is applied in combination with the knn and Svm classifiers. In these cases this method has a considerable positive impact and knn and Svm achieve their best performance. (e) The performance of *CSR* is also influenced by the availability of words in the vicinity of concepts in the source and target ontologies. Specifically, when few words (no comments, no instances or no defined properties) are available, then its discriminating power, and thus its performance, drops. This is especially the case when the method exploits latent features, where many words are necessary for effective inference.

Our future work mainly includes (a) the synthesis of *CSR* with other methods in order to refine and improve the achieved performance of synthesized methods, (b) the investigation of other kinds of classification features and dataset balancing methods, (c) the investigation of incremental machine learning algorithms in order to store and refine the learned patterns in the form of background knowledge, and (d) the adaption of the *CSR* method to locate more types of non-equivalence relations (e.g. disjointness).

Acknowledgments

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

References

[1] J. Euzenat, P. Shvaiko, Ontology Matching, Springer-Verlag New York, Inc, Secaucus, NJ, 2007.
[2] O. Svab, V. Svatek, H. Stuckenschmidt, A Study in Empirical and 'Casuistic' Analysis of Ontology Mapping Results, ESWC, Innsbruck, Austria, 2007, pp. 655–669.
[3] T. Mitchell, Concepts learning and the general-to-specific ordering, in: T. Mitchell (Ed.), Machine Learning, The McGraw-Hill Companies, Inc., 1997, pp. 20–51.
[4] V. Spiliopoulos, A. Valarakos, G. Vouros, CSR: Discovering Subsumption Relations for the Alignment of Ontologies, in: European Semantic Web Conference, Tenerife, Spain, 2008, pp. 418–431.
[5] Y. Kalfoglou, M. Schorlemmer, Ontology mapping: The state of the art, The Knowledge Engineering Review Journal 18 (1) (2003) 1–31.
[6] K. Kotis, G. Vouros, K. Stergiou, Towards Automatic Merging of Domain Ontologies: The HCONE-merge approach, Journal of Web Semantics 4 (1) (2006) 60–79.
[7] OWL Web Ontology Language Overview, W3C Recommendation 10 February 2004, http://www.w3.org/TR/owl-features/.
[8] I. Horrocks, P.F. Patel-Schneider. Reducing OWL Entailment to Description Logic Satisfiability, 2003, 17–29.
[9] F. Giunchiglia, M. Yatskevich, P. Shvaiko, Semantic Matching: Algorithms and implementation, Journal on Data Semantics IX (2007) 1–38.
[10] P. Bouquet, L. Serafini, S. Zanobini, S. Sceffer, Bootstrapping semantics on the web: meaning elicitation from schemas, WWW, Edinburgh, Scotland, 2006, pp. 505–512.
[11] Z. Aleksovski, M. Klein, W. Kate, F. Harmelen, Matching Unstructured Vocabularies Using a Background Ontology, EKAW, Podebrady, Czech Republic, 2006, pp. 182–197.
[12] J. Gracia, V. Lopez, M. D'Aquin, M. Sabou, E. Motta, E. Mena, Solving Semantic Ambiguity to Improve Semantic Web based Ontology Matching Ontology Matching Workshop, Busan Korea, 2007.
[13] V. Lopez, M. Sabou, E. Motta, Powermap: Mapping the Real Semantic Web on the Fly, ISWC, Athens, GA, USA, 2006, pp. 414–427.
[14] R. Gligorov, W. ten Kate, Z. Aleksovski, F. van Harmelen, Using Google Distance to Weight Approximate Ontology Matches, WWW, Banff, Alberta, Canada, 2007, pp. 767–776.
[15] W.R. Van Hage, S. Katrenko, G. Schreiber, A Method to Combine Linguistic Ontology Mapping Techniques, ISWC, Osaka, Japan, 2005, pp. 732–744.
[16] P. Cimiano, S. Staab, Learning by googling, ACM SIGKDD Explorations Newsletter. 6:2, 2004, pp. 24–33.
[17] M. Hearst, Automatic acquisition of hyponyms from large text corpora International Conference on Computational Linguistics, Nantes France, 1992, pp. 539–545.
[18] F. Hamdi, H. Zargayouna, B. Safar, C. Reynaud, TaxoMap in the OAEI 2008 alignment contest, Ontology Alignment Evaluation Initiative Campaign, Ontology Matching Workshop, Karlsruhe, Germany, 2008.
[19] J. David, F. Guillet, H. Briand, An interactive, asymmetric and extensional method for matching conceptual hierarchies, EMOI-INTEROP Workshop, Luxembourg, 2006.
[20] H. Nottelmann, U. Straccia, A Probabilistic, Logic-based Framework for Automated Web Directory Alignment, Soft Computing in Ontologies and the Semantic Web Series: Studies in Fuzziness and Soft Computing Series, Zongmin Ma, ed. Springer Verlag, 2006.
[21] G.H. John, P. Langley, Estimating Continuous Distributions in Bayesian Classifiers, in: Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Mateo, 1995, pp. 338–345.
[22] D. Aha, D. Kibler, Instance-based learning algorithms, Machine Learning 6 (1991) 37–66.
[23] C.C. Chang, C.J. Lin. LIBSVM: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001.
[24] T. Mitchell, Decision tree learning, in: T. Mitchell (Ed.), Machine Learning, The McGraw-Hill Companies, Inc, 1997, pp. 52–78.
[25] N. Japkowicz, S. Stephen, The class imbalance problem: a systematic study, IDA Journal 6 (5) (2002) 429–449.
[26] G.E.A.P.A. Batista, R.C. Prati, M.C. Monard. A study of the behavior of several methods for balancing machine learning training data, ACM SIGKDD Explorations Newsletter, 6:1, 2004, pp. 20–29.
[27] M. Griffiths, Steyvers A probabilistic approach to semantic representation, in: Annual Conference of the Cognitive Science Society, Fairfax, Virginia, US, 2002, pp. 381–386.
[28] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, Journal of Machine Learning Research 3 (2003) 993–1022.
[29] V. Spiliopoulos, A.G. Valarakos, G.A. Vouros, V. Karkaletsis, SEMA: Results for the ontology alignment contest OAEI 2007, Ontology Alignment Evaluation Initiative Campaign, Ontology Matching Workshop, Busan, Korea, 2007.
[30] G. Salton, M.H. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, NY, USA, 1986.
[31] Ontology Alignment Evaluation Initiative, http://oaei.ontologymatching.org/.
[32] Illinois Semantic Web Archive, http://anhai.cs.uiuc.edu/archive/.
[33] Consensus Workshop Track, OAEI 2006, http://oaei.ontologymatching.org/2006/conference/.
[34] O. Svab, V. Svatek, P. Berka, D. Rak, P. Tomasek, OntoFarm: Towards an Experimental Collection of Parallel Ontologies, Poster Track of ISWC, Galway, Ireland, 2005.
[35] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA, 1993.
[36] T. Fawcett, An introduction to ROC analysis, Pattern Recognition Letters 27 (8) (2006) 861–874.

Please cite this article in press as: V. Spiliopoulos, et al., On the discovery of subsumption relations for the alignment of ontologies, Web Semantics: Sci. Serv. Agents World Wide Web (2010), doi:10.1016/j.websem.2010.01.001