Gold Standard Based Ontology Evaluation Using Instance Assignment

Janez Brank
Jozef Stefan Institute
Jamova 39
Ljubljana, Slovenia
+386 1 477 3778
janez.brank@ijs.si

Dunja Mladenić
Jozef Stefan Institute
Jamova 39
Ljubljana, Slovenia
+386 1 477 3772
dunja.mladenic@ijs.si

Marko Grobelnik
Jozef Stefan Institute
Jamova 39
Ljubljana, Slovenia
+386 1 477 3778
marko.grobelnik@ijs.si

ABSTRACT

An ontology is an explicit formal conceptualization of some domain of interest. Ontology evaluation is the problem of assessing a given ontology from the point of view of a particular criterion or application, typically in order to determine which of several ontologies would best suit a particular purpose. This paper proposes an ontology evaluation approach based on comparing an ontology to a gold standard ontology, assuming that both ontologies are constructed over the same set of instances.

1. Introduction

Different knowledge discovery methods have been adopted for the problem of semi-automated ontology construction (GROBELNIK AND MLADENIC, 2005) including: unsupervised, semi-supervised and supervised learning over a collection of text documents; using natural language processing to obtain a semantic graph of a document; visualization of documents; information extraction to find relevant concepts; and visualization of the context of named entities in a document collection.

Users facing a multitude of ontologies need to have a way of assessing them and deciding which one best fits their requirements. Likewise, people constructing an ontology need a way to evaluate the resulting ontology and possibly to guide the construction process and any refinement steps. Automated or semi-automated ontology learning techniques also require effective evaluation measures, which can be used to select the "best" ontology out of many candidates, to select values of tunable parameters of the learning algorithm, or to direct the learning process itself if the latter is formulated as finding a path through a search space.

The remainder of this Chapter is structured as follows. In Section 2, we present related work on ontology evaluation. In Section 3, we refer to a formal framework for defining an ontology and show how various aspects of evaluation can be incorporated in such a framework. In Section 4, we present our approach to evaluating a hierarchic ontology by comparing it to a "gold standard". In Section 5, we test this approach on a real-world topic ontology. In Section 6, we present some guidelines for future work.

2. Related Work

Various approaches to the evaluation of ontologies have been considered in the literature, depending on what kind of ontologies are being evaluated and for what purpose. Broadly speaking, most evaluation approaches fall into one of the following categories:

- approaches based on comparing the ontology to a "gold standard" (which may itself be an ontology; e.g. MAEDCHE AND STAAB, 2002);
- approaches based on using the ontology in an application and evaluating the results (e.g. PORZEL AND MALAKA, 2004);
- approaches involving comparisons with a source of data (e.g. a collection of documents) about the domain that is to be covered by the ontology (e.g. BREWSTER *et al*., 2004);
- approaches where evaluation is done by humans who try to assess how well the ontology meets a set of predefined criteria, standards, requirements, etc. (e.g. LOZANO-TELLO AND GÓMEZ-PÉREZ, 2004).

In addition to the above categories of evaluation, we can group the ontology evaluation approaches based on the level of evaluation, as described in BRANK *et al.* (2006).

3. A Theoretical Framework for Ontology Evaluation

A reasonable and well thought-out formal definition of ontologies has been described recently in the work of EHRIG *et al.* (2005). In this formalization, the ontology (with datatypes) is defined as a structure $O = (C, T, R, A, I, V, \leq_C, \leq_T, \sigma_R, \sigma_A, \iota_C, \iota_T, \iota_R, \iota_A)$. It consists of (disjoint) sets of concepts (C), types (T), relations (R), attributes (A), instances (I), and values (V). The partial orders $\leq_C$ (on C) and $\leq_T$ (on T) define a concept hierarchy and a type hierarchy. The function σ_R: $R \to C^2$ provides relation signatures (i.e. for each relation, the function specifies which concepts may be linked by this relation), while σ_A: $A \to C \times T$ provides attribute signatures (for each attribute, the function specifies to which concept the attribute belongs and what is its datatype). Finally, there are partial instantiation functions ι_C: $C \to 2^I$ (the assignment of instances to concepts), ι_T: $T \to 2^V$ (the assignment of values to types), ι_R: $R \to 2^{I \times I}$ (which instances are related by a particular relation), and ι_A: $A \to 2^{I \times V}$ (what is the value of each attribute for each instance). (Another formalization of ontologies, based on similar principles, has also been described by BLOEHDORN *et al.* (2005)).

For some types of ontologies, this framework can be further extended, particularly with "concept attributes" in addition to the "instance attributes" mentioned above. The concept attributes would be a set A', with a signature function $\sigma_{A'}$: $A' \to T$ and an instantiation function $\iota_{A'}$: $A' \to 2^{C \times V}$. The value of such an attribute would not be associated to a particular instance of a concept, but would apply to the concept as such. This extension will be useful for some of the evaluation scenarios considered later in this section. Other possible extensions, such as relations between concepts (as opposed to between instances), the introduction of

metaclasses, or the introduction of relations with arity greater than 2, are probably of less practical interest.

A flexible formal network like this can accommodate various commonly-used kinds of ontologies:

- *Terminological ontologies* where concepts are word senses and instances are words, e.g. the WordNet ontology. Attributes include things like natural-language descriptions of word senses (for concepts) and string representations of words (for instances).
- *Topic ontologies* where concepts are topics and instances are documents. Familiar examples include the Open Directory (dmoz.org) or the Yahoo! directory. Concept attributes typically consist of a name and a short description of each topic, and instance attributes consist of a document title, description, URL, and the main block of the text (for practical purposes, such text is often represented as a vector using e.g. the TF-IDF weighting under the vector space model of text representation).
- *Data-model ontologies* where concepts are tables in a data base and instances are data records (such as in a database schema). In this setting, datatypes and attributes in the above-mentioned formal definition of an ontology are straightforward analogies to the types and attributes (a.k.a. fields or columns) in a data base management system.

Evaluation can be incorporated in this theoretical framework as a function that maps the ontology O to a real number, e.g. in the range [0, 1]. However, a more practical approach is to focus the evaluation on individual components of the ontology O (which correspond roughly to different levels of ontology evaluation; BRANK *et al.*, 2006). Results of the evaluation of individual components can later be aggregated into a combined ontology evaluation score (EHRIG *et al.*, 2005).

- The datatypes and their values (i.e. T, V, $\leq_T$, and ι_T) would typically not be evaluated; they are merely the groundwork on which the rest of the structure can stand.
- A lexical- or concept-level evaluation can focus on C, I, ι_C, and possibly some instance attributes from ι_A.
- Evaluation of the concept hierarchy (is-a relationship) would focus on the $\leq_C$ partial order.
- Evaluation of other semantic relations would focus on R, ι_R, and the concept and instance attributes.
- One could also envision evaluation focusing on particular attributes; for example, whether a suitable natural-language name has been chosen for each concept. This kind of evaluation would take ι_C and the attributes as input and assess whether the concept attributes are suitable given ι_C and the instance attributes.
- Application- or task-based evaluation could be formalized by defining the application as a function $A(D, O)$ which produces some output given its input data D and the ontology O. By fixing the input data D, any evaluation function defined on the outputs of A becomes de facto an evaluation function on O. However, the practical applicability of such a formalization is debatable.
- Evaluation based on comparison to a gold standard can be incorporated into this theoretical framework as a function defined on a pair of ontologies (effectively a kind of similarity measure, or a distance function between ontologies). Similarly, data-driven evaluation can be seen as a function of the ontology and the domain-specific data corpus D, and could even be formulated probabilistically as $P(O|D)$.

4. Architecture and Approach

We have developed an approach to ontology evaluation primarily geared to enable automatic evaluation of an ontology that includes instances of the ontology concepts. The approach is based on the gold standard paradigm and its main focus is to compare how well the given ontology resembles the gold standard in the arrangement of instances into concepts and the hierarchical arrangement of the concepts themselves. It is similar to the other existing ontology evaluation methods based on the gold standard (see Section 2) with a main difference in basing the evaluation on instances assigned to the ontology concepts: our approach does not rely on natural-language descriptions of the concepts and instances (unlike e.g., the string edit distance approaches of MAEDCHE AND STAAB, 2002). No assumptions are made regarding the representation of instances, only that we can distinguish one instance from another (and that the ontology is based on the same set of instances as the gold standard).

4.1 Similarity measures on partitions

Our approach to evaluation is based on the analogies between this ontology learning task and traditional unsupervised clustering. In clustering, the task is to partition a set of instances into a family of disjoint subsets. Here, the topic ontology can be seen as a hierarchical way of partitioning the set of instances. The clustering community has proposed various techniques for comparing two partitions of the same set of instances, which can be used to compare the output of an automated clustering method with a gold-standard partition. If these distance measures on traditional "flat" partitions can be extended to hierarchical partitions, they can be used to compare a learned ontology to the gold-standard ontology (since both will be, in the context of this ontology learning task, two hierarchical partitions of the same set of instances).

One popular measure of agreement between two flat partitions is the Rand index (RAND, 1971). Assume that there is a set of instances $O = \{o_1, ..., o_n\}$ (in this section we will denote the set of instances by O rather than I as in the previous section, to prevent confusion with i as an index in subscripts), with two partitions of O into a family of disjoint subsets, $U=\{U_1, ..., U_m\}$ and $V=\{V_1, ..., V_k\}$, where $\cup_{i=1..m} U_i = O$, $\cup_{j=1..k} V_j = O$, $U_i \cap U_{i'} = \{\}$ for each $1 \leq i < i' \leq m$, and $U_j \cap U_{j'} = \{\}$ for each $1 \leq j < j' \leq k$. Then one way to compare the partitions U and V is to count the agreements and disagreements in the placement of instances into clusters. If two items o_i, $o_j \in O$ belong to the same cluster of U but to two separate clusters of V, or vice versa, this is considered a disagreement. On the other hand, if they belong to the same cluster in both partitions, or to separate clusters in both partitions, this is considered an agreement between partitions. The Rand index between U and V is the number of agreements relative to the total number of pairs of instances (i.e. to $n(n-1)/2$).

4.2 A similarity measure for ontologies

We can elegantly formulate a similarity measure over ontologies by rephrasing the Rand index as follows. Let us denote by $U(o)$ the cluster of U that contains the instance $o \in O$, and similarly by $V(o)$ the cluster of V that contains the instance $o \in O$. Let $\delta_x(X_i, X_j)$ be some distance measure between clusters X_i and X_j. Then we define the OntoRand index by the following formula:

$$OntoRandIdx(U, V) = 1 - [\Sigma_{1 \leq i < j \leq n} |\delta_U(U(o_i), U(o_j)) - \delta_V(V(o_i), V(o_j))|] / [n(n-1)/2]. \quad (1)$$

If we define $\delta_U(U_i, U_j) = 1$ if $U_i = U_j$, and $\delta_U(U_i, U_j) = 0$ otherwise and δ_V as well in an analogous manner, we can see that the Rand index is a special case of our OntoRand index. That is, the term bracketed by |...| in eq. (1) equals 1 if there is a disagreement between U and V concerning the placement of the pair of instances o_i and o_j. The sum over all i and j therefore counts the number of pairs where a disagreement occurs.

When we apply the OntoRand index for the purpose of comparing ontologies, we must take the hierarchical arrangement of concepts into account. In the original Rand index, what matters for a particular pair of instances is simply if they belong to the same cluster or not. However, when concepts or clusters are organized hierarchically, not any two different clusters are equally different. For example, two concepts with a common parent in the tree are likely to be quite similar even though they are not exactly the same; on the other hand, two concepts that do not have any common ancestor except the root of the tree are probably highly unrelated. Thus, if one ontology places a pair of instances in the same concept while the other ontology places this pair of instances in two different concepts with a common parent, this is a disagreement, but not a very strong one; on the other hand, if the second ontology places the two instances into two completely unrelated concepts, this would be a large disagreement. We use the formula for *OntoRandIdx*(U, V) given above, where the functions δ_U and δ_V take this intuition into account. That is, rather than returning merely 1 or 0 depending on whether the given two clusters are the same or not, the functions δ_U and δ_V should return a real number from the range [0, 1], expressing a measure of how closely related the two clusters are.

By plugging in various definitions of the functions δ_U and δ_V, we can obtain a family of similarity measures for ontologies, suitable for comparing an ontology with the gold standard in the context of the task that has been discussed in Section 4.1. We propose two concrete families of δ_U and δ_V. Since the definitions of δ_U and δ_V will always be analogous to each other and differ only in the fact that each applies to a different ontology, we refer only to the δ_U function in the following discussion.

4.2.1 Similarity based on common ancestors

One possibility is inspired by the approach that is sometimes used to evaluate the performance of classification models for classification in hierarchies (see e.g. MLADENIĆ, 1998), and that could incidentally also be useful in the context of e.g. evaluating an automatic ontology population system. Given a concept U_i in the ontology U, let $A(U_i)$ be the set of all ancestors of this concept, i.e. all concepts on the path from the root to U_i (including U_i itself). If two concepts U_i and U_j have a common parent, the sets $A(U_i)$ and $A(U_j)$ will have a large intersection; on the other hand, if they have no common parent except the root, the intersection of $A(U_i)$ and $A(U_j)$ will contain only the root concept. Thus the size of the intersection can be taken as a measure of how closely related the two concepts are.

$$\delta_U(U_i, U_j) = |A(U_i) \cap A(U_j)| \,/\, |A(U_i) \cup A(U_j)|. \qquad (2)$$

This measure has the additional nice characteristic that it can be extended to cases where U is not a tree but an arbitrary directed acyclic graph. If the arrows in this graph point from parents to children, the set $A(U_i)$ is simply the set of all nodes from which U is reachable.

4.2.2 Similarity based on distance in the tree

An alternative way to define a suitable function δ_U would be to work directly with the distances between U_i and U_j in the tree U. In this case, let l be the distance between U_i and U_j in the tree (length of the path from U_i to the common ancestor of U_i and U_j, and thence down to U_j), and h be the depth of the deepest common ancestor of U_i and U_j. If l is large, this is a sign that U_i and U_j are not very closely related; similarly, if h is small, this is a sign that U_i and U_j don't have any common ancestors except very general concepts close to the root, and therefore U_i and U_j aren't very closely related. There are various ways of taking these intuitions into account in a formula for δ_U as a function of l and h. For example, RADA *et al.* (1989) have proposed a distance measure of the form:

$$\delta(l, h) = e^{-\alpha l} \operatorname{th}(\beta h) \qquad (3)$$

Here, α and β are nonnegative constants, and th is the hyperbolic tangent

$$\operatorname{th}(x) = (e^x - e^{-x}) \,/\, (e^x + e^{-x}) = 1 - 2/(1 + e^{2x}).$$

Thus, if h is small, $\operatorname{th}(\beta h)$ is close to 0, whereas for a large h it becomes close to 1. It is reasonable to treat the case when the two concepts are the same, i.e. when $U_i = U_j$ and thus $l = 0$, as a special case, and define $\delta(0, h) = 1$ in that case, to prevent $\delta_U(U_i, U_i)$ from being dependent on the depth of the concept U_i.

Incidentally, if we set α to 0 (or close to 0) and β to some large value, $\delta(l, h)$ will be approx. 0 for h = 0 and approx. 1 for h > 0. Thus, in the sum used to define the OntoRand index (1), each pair of instances contributes the value of 1 if they have some common ancestor besides the root in one ontology but not in other, otherwise it contributes the value of 0. Thus, the OntoRand index becomes equivalent to the ordinary Rand index computed over the partitions of instances implied by the second-level concepts of the two ontologies (i.e. the immediate subconcepts of the root concept). This can be taken as a warning that α should not be too small and β not too large, otherwise the OntoRand index will ignore the structure of the lower levels of the ontologies.

The overlap-based version of d_U from eq. (2) can also be defined in terms of h and l. If the root is taken to be at depth 0, then the intersection of $A(U_i)$ and $A(U_j)$ contains $h + 1$ concepts, and the union of $A(U_i)$ and $A(U_j)$ contains $h + l - 1$ concepts. Thus, we see that eq. (2) is equivalent to defining

$$\delta(l, h) = (h + 1) \,/\, (h + l + 1). \qquad (4)$$

By comparing the equations (3) and (4), we see a notable difference between the two definitions of δ: when $h = 0$, i.e. when the two instances have no common ancestor except the root, eq. (3) returns $\delta = 0$ while eq. (4) returns $\delta = 1 / (l + 1) > 0$. When comparing two ontologies, it may often happen that many pairs of instances have no common ancestor (except the root) in either of the two ontologies, i.e. $h_U = h_V = 0$, but the distance between their concepts is likely to be different: $l_U \neq l_V$. In these cases, using eq. (3) will result in $\delta_U = \delta_V = 0$, while eq. (4) will result in $\delta_U \neq \delta_V$. When the resulting values $|\delta_U - \delta_V|$ are used in eq. (1), we see that in the case of definition (4), many terms in the sum will be 0 and the OntoRand index will be close to 1. For example, in our experiments with the Science subtree of dmoz.org (Sec. 5.3), despite the fact that the assignment of instances to concepts was considerably different between the two ontologies, approx. 81% of instance pairs had $h_U = h_V = 0$ (and only 3.2% of these additionally had $l_U = l_V$). Thus, when using the definition of δ from eq. (3) (as opposed to the overlap-based definition from eq.

(4)), we must accept the fact that most of the terms in the sum (1) will be 0 and OntoRand index will be close to 1. This does not mean that the resulting values of OntoRand are not useful for assessing whether e.g. one ontology is closer to the gold standard than another ontology is, but it may nevertheless appear confusing that OntoRand is always so close to 1. In this case a possible alternative is to replace eq. (3) by

$$\delta(l, h) = e^{-\alpha l} \operatorname{th}(\beta(h + 1)) \qquad (3')$$

The family of δ-functions defined by (3') can be seen as a generalization (in a loose sense) of the δ-function from formula (4). For example, we compared the values of δ produced by these two definitions on a set of 10^6 random pairs of documents from the dmoz.org Science subtree. For a suitable choice of α and β, the definition (3') can be made to produce values of δ that are very closely correlated with those of definition (4) (e.g. correl. coefficient = 0.995 for α = 0.15, β = 0.25). Similarly, when we compute $|\delta_U - \delta_V|$ for various pairs of documents (when using eq. (1) to compare two ontologies in Sec. 5.3), definition (3') can yield values closely correlated to those of definition (4) for suitable values of α and β (e.g. correl. coef. = 0.981 for α = 0.8, β = 1.5). However, note that the fact that δ values of (3') are closely correlated with those of (4) for some choice of α and β does not imply that the $|\delta_U - \delta_V|$ will also be closely correlated for the *same* choice of α and β (or vice versa).

The need to select concrete values of α and β is one of the disadvantages of using the definition (3) (or (3')) rather than the overlap-based definition (2) (or equivalently (4)).

Further generalizations. The distance measure (3) could be further generalized by taking $\delta(l, h) = f(l)\, g(h)$ for any decreasing function f and increasing function g. Since the values l and h are always integers and are limited by the depth of the tree (or twice the depth in the case of l), the functions f and g (or even $\delta(l, h)$ itself) could even be defined using a table of function values for all possible l and h.

Note that the main part of the OntoRand index formula, as defined in equation (1), i.e. the sum $\Sigma_{1 \le i < j \le n} |\delta_U(U(o_i), U(o_j)) - \delta_V(V(o_i), V(o_j))|$, can also be interpreted as a Manhattan (L_1-norm) distance between two vectors of $n(n-1)/2$ components, one depending on the ontology U and the other depending only on the ontology V. Thus, in effect, we have represented an ontology U by a "feature vector" in which the (i, j)-th component has the value $\delta_U(U(o_i), U(o_j))$ describing how closely the instances o_i and o_j have been placed in that ontology. This interpretation opens the possibility of various further generalizations, such as using Euclidean distance instead of Manhattan distance, or even using kernel methods (cf. HAUSSLER, 1999). However, we leave such extensions for further work.

4.3 Approximation algorithms

As can be seen from eq. (1), the computation of our ontology similarity measure involves a sum over all pairs of documents, (i, j) for $1 \le i < j \le n$. This quadratic time complexity can be problematic when comparing ontologies with a fairly large number of instances (e.g. on the order of 100000, as in the case of the dmoz.org "Science" subtree mentioned in Section 5). One way to speed up the computation of the similarity measure and obtain an approximate result is to use a randomly sampled subset of pairs rather than all possible pairs of documents. That is, eq. (1) would then contain the average value of $|\delta_U(U(o_i), U(o_j)) - \delta_V(V(o_i), V(o_j))|$ over some subset of pairs instead of over all pairs.

Another way towards approximate computation of the similarity measure is to try to identify pairs (i, j) for which the difference $|\delta_U(U(o_i), U(o_j)) - \delta_V(V(o_i), V(o_j))|$ is not close to 0. If both ontologies classify the instances o_i and o_j into highly unrelated clusters, the values $\delta_U(U(o_i), U(o_j))$ and $\delta_V(V(o_i), V(o_j))$ will both be close to 0 and their difference will also be close to 0 and will not have a large effect on the sum. (In a typical dmoz-like hierarchy we can expect that a large proportion of pairs of instances will fall unto such relatively unrelated clusters. As an extreme case, consider the definition of δ_U using eq. (3). If a pair of instances has no common ancestor concept except the root, h will be 0 and thus δ_U will be 0. If this happens in both ontologies, the pair will contribute nothing to the sum in eq. (1).) Thus it would be reasonable to try identifying pairs (i, j) for which o_i and o_j are in closely related clusters in at least one of the two ontologies, and computing the exact sum for these pairs, while disregarding the remaining pairs (or processing them using the subsampling technique from the previous paragraph). For example, suppose that δ_U is defined by eq. (4) as $\delta(l, h) = (h + 1) / (h + l + 1)$. Thus, we need to find pairs of concepts for which $(h + 1) / (h + l + 1)$ is greater than some threshold ε. (Then we will know that detailed processing is advisable for pairs of instances which fall into one of these pairs of concepts.) The condition $(h + 1) / (h + l + 1) > \varepsilon$ can be rewritten as $l < (h + 1)(1/\varepsilon - 1)$. Thus, suitable pairs of concepts could be identified by the following algorithm:

Initialize $P := \{\}$.
For each concept c:
  Let h be the depth of c, and let
    $L = \lfloor (h + 1)(1/\varepsilon - 1) \rfloor$.
  Denote the children of c (its immediate subconcepts)
    by $c_1, \ldots, c_r$.
  For each l from 1 do L, for each i from 1 to r,
    let $S_{l,i}$ be the set of those subconcepts of c that
    are also subconcepts of c_i and are l levels
    below c in the tree.
  For each l from 1 to L, for each i from 1 to r,
    add to P all the pairs from
    $S_{l,i} \times (\cup_{l' \le L-1} \cup_{i' \ne i} S_{l',i'})$.

In each iteration of the outermost loop, the algorithm processes a concept c and discovers all pairs of concepts c', c'' such that c is the deepest common ancestor of c' and c'' and $\delta_U(c', c'') > \varepsilon$. For more efficient maintenance of the $S_{l,i}$ sets, it might be advisable to process the concepts c in a bottom-up manner, since the sets for a parent concept can be obtained by merging appropriate sets of its children.

For the time being, we have tested random sampling of pairs as outlined at the beginning of this Subsection. Separate treatment of pairs with $(h + 1) / (h + l + 1) > \varepsilon$ will be the topic of future work.

5. Evaluation of the proposed approach

The idea of evaluating the proposed approach to automatic ontology evaluation is in showing its output on several concrete situations enabling the reader to get an idea of the approach results given a well defined mismatch in the ontologies (the learned ontology and the "gold-standard" ontology). Namely, instead of learning and ontology that we then evaluate, we use the "gold-standard" ontology, introduce some errors in it and use it to simulate the learned ontology. We have defined several simple and intuitive operations for introducing errors in the "gold-

standard" ontology. The aim is to illustrate a kind of mismatch that can be found between the learned ontology and the "gold-standard" ontology and its influence on the evaluation score of the proposed OntoRand index. The following operations are presented below in our evaluation of the proposed approach:

- Removing lower levels of the tree – deleting all concepts below a certain depth in the tree (see Section 5.1).
- Swapping a concept and its parent (see Section 5.2).
- Reassigning instances to concepts based on their associated natural language text (see Section 5.3).

We have tested our approach on a concrete task of evaluating a topic ontology based on the dmoz.org internet directory. This ontology is structured as a hierarchy of topics, and each topic may contain (besides subtopics) zero or more links to external web pages.

We used a version of the dmoz.org directory downloaded on October 21, 2005. It contains 687,333 concepts and 4,381,225 instances. The concepts are organized into a tree; the deepest parts of the hierarchy go 15 levels deep, but in most places it is shallower (85% of all concepts are on levels 5 through 9, and the average node depth is 7.13). Since it would be too time-consuming to compute our OntoRand index over all pairs of documents (there are approx. $9.6 \cdot 10^{12}$ such pairs), we used a random sample of 10^6 pairs of documents.

In the case of the similarity measure (3), which is based on the tree distance between two concepts, it is necessary to select the parameters α and β. Recall that α is used in the term $e^{-\alpha l}$, where l is the length of the path from one concept to the other. Since our hierarchy has just 15 levels, we know that $l \leq 28$ for any pair of nodes; but since most nodes are on levels 5 through 9, we can expect l to be around 10–15 for a typical random pair of unrelated concepts. We decided to use $\alpha = 0.3$, which results in $e^{-\alpha l}$ values from 0.74 (for $l = 1$) to 0.22 (for $l = 5$), 0.05 (for $l = 10$) and 0.01 (for $l = 15$).

The parameter β can be chosen using similar considerations. It is used in the term $\mathrm{th}(\beta h)$, where h is the level at which the last common ancestor of the two concepts is located. Thus in our case h will be between 0 and 14, and will be close to 0 for two random unrelated concepts. For two very closely related concepts, h will typically be close to the depth of these two concepts, which (as we saw above) is on average around 7. We use $\beta = 0.4$, which results in values of $\mathrm{th}(\beta h)$ ranging from 0 (for $h = 0$) and 0.20 (for $h = 1$) to 0.76 (for $h = 5$), 0.89 (for $h = 7$), and 0.96 (for $h = 10$).

In general, the choice values of α and β depends on the characteristics of the ontologies we are dealing with. A more principled way of choosing α and β might be to set explicit requirements on the value that we want $e^{-\alpha l}$ to have for a pair of two random (i.e. typically unrelated) documents, and on the value that we want $\mathrm{th}(\beta h)$ to have for a pair of two very closely related documents.

5.1 Removing lower levels of the tree

In this scenario we keep only the upper k levels of the tree, for various values of k. Any concepts located at levels from k+1 on are discarded; instances that used to be assigned to one of the deleted concepts are reassigned to its ancestor on the level k–1 (i.e. the deepest level that was not deleted). We then compare the resulting tree with the original tree. This removal of lower levels of the tree corresponds to the scenario that the ontology is being constructed automatically in a top-down manner (e.g. by hierarchical top-down clustering of instances) and some automatic stopping criterion is used to decide when to stop partitioning the clusters; if we stop too early, the resulting hierarchy will lack the lower levels. The chart in Figure 1 shows how the overlap measure (eq. 2) and the tree distance measure (eq. 3) react to this gradual removal of lower parts of the hierarchy.

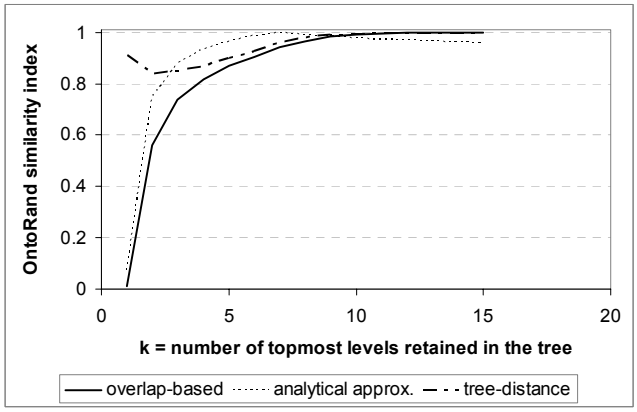

Fig. 1. Evaluation of ontologies that lack lower levels, based on the OntoRand index. The overlap-based similarity measure uses formula (2) to define δ_U, while the tree-distance based similarity measure uses formula (3). The dotted line shows an analytical approximation of the OntoRand values based on the overlap similarity measure.

We note that the **overlap-based similarity measure** increases monotonically as more and more levels are kept. The increase is quick at first and slow at the end, which is reasonable because (as has been noted above) the deeper levels of the hierarchy contain relatively few nodes, so discarding them does not alter the hierarchy so dramatically. For instance, if we constructed an ontology in a top-down manner and stopped when the ontology is at most seven levels deep, the OntoRand index would estimate the similarity of this ontology to the gold standard (having an average node depth of approx. 7) as 0.94. On the other hand, if we stopped after at most three levels, the OntoRand index would be 0.74.

It may be somewhat surprising that the similarity of an ontology to the original one is still as high as 0.74 even if only the top three levels of the ontology have been kept. To understand this, consider a pair of random concepts; in the original hierarchy, they are typically unrelated and are located around the 7th level, so the ancestor sets of eq. (2) have an intersection of 1 and a union of around 13, resulting in the overlap measure $\delta \approx 1/13$. In the pruned hierarchy, where only k uppermost levels have been retained, and documents from lower nodes reassigned to the ancestor nodes at level k–1, such a random pair of documents would yield δ around $1/(2k–1)$. Thus such pairs of documents would push the OntoRand index value towards $1 - |{}^{1}/_{13} - {}^{1}/_{(2k-1)}|$. As the "analytical approximation" in the chart shows, this formula is not an altogether bad predictor of the shape of the curve for the overlap-based measure.

The **tree-distance similarity measure** is slightly more problematic in this scenario. In the original tree, a typical random pair of instances falls into unrelated concepts that have no common ancestors except the root, i.e. $h = 0$ and thus $\delta = 0$ (or δ close to 0 even if $h > 0$). If a few deepest levels of the tree are removed and instances reassigned to the suitable ancestor concepts, any pair of instances that used to have $h = 0$ will still have $h = 0$, thus its δ according to eq. (3) remains unchanged and this pair does not help decrease the similarity measure between the new hierarchy

and the original one. This is why the similarity as measured by OntoRand remains relatively high all the time. Only concept pairs with $h > 0$ contribute towards the dissimilarity, because their distance (l in eq. (3)) decreases if the lower levels are pruned away and the instances moved to higher-level concepts. Because l is used in the term $e^{-\alpha l}$, decreasing l causes the value of δ to increase for that pair of instances; the more levels we prune away, the larger δ will be compared to its original value, and the OntoRand similarity decreases accordingly. A quirk occurs at the very end, when only one level remains and h drops to 0 even for these pairs of instances; thus δ doesn't increase when we move from two levels to 1: it drops to 0 instead, causing the overall OntoRand similarity to grow again. This non-monotonicity could be addressed by modifying the formula (3) somewhat, but it doesn't really have a large practical impact anyway, as in a practical setting the ontology to be compared to the gold standard would certainly have more than one level.

5.2 Swapping a concept and its parent

This operation on trees is sometimes known as "rotation". Consider a concept c and its parent concept c'. This operation replaces c and c' so that c' becomes the child of c; all other children of c', which were formerly the siblings of c, are now its grandchildren; all the children of c, which were formerly the grandchildren of c', are now its siblings. If c' formerly had a parent c'', then c'' is now the parent of c, not of c'. The result of this operation is a tree such as might be obtained by an automated ontology construction algorithm that proceeds in a top-down fashion and did not split the set of instances correctly (e.g. instead of splitting the set of instances related to science into those related to physics, chemistry, biology, etc., and then splitting the "physics" cluster into mechanics, thermodynamics, nuclear physics, etc., it might have split the "science" cluster into mechanics, thermodynamics, nuclear physics, and "miscellaneous", where the last group would later be split into chemistry, biology, etc.).

How does this operation affect the values of h and l used in eqs. (2) and (3)? For two concepts that were originally both in the subtree rooted by c, the value of h decreases by 1; if they were both in the subtree of c' but not in the subtree of c, the value of h increases by 1; if one was in the subtree of c and the other outside the subtree of c', the value of l decreases by 1; if one was in the subtree of c' but not in the subtree of c, and the other was outside the subtree of c', the value of l increases by 1; otherwise, nothing changes. The last case includes in particular all those pairs of instances where none belonged to the subtree rooted by c' in the original ontology; this means the vast majority of pairs (unless the subtree of c' was very large). Thus the disagreement in the placement of documents is usually quite small for an operation of this type, and OntoRand is close to 1. This phenomenon is even more pronounced when using the similarity measure based on tree distance (eq. 3) instead of the overlap measure (eq. 2). Therefore, in Figure 2, we show only the results for the overlap measure and we show 1–OntoRand instead of OntoRand itself.

We performed 640 experiments with this operation, using each of the 640 third-level categories as the category c (e.g. replacing Top/Science/Physics and Top/Science, etc.). Experiments show that the dissimilarity of the ontology after rotation to the original ontology grows with the size of the parent subtree of c, while this dissimilarity decreases with the size of c's own subtree. This is reasonable: the more instances there are in c's subtree, the less different it is from its parent, and the less the ontology has changed due to the rotation. For instance, the topmost group of "×" symbols on Fig. 2 corresponds to experiments where c was one of the subcategories of the largest second-level category, "Top/World". As this chart shows, the dissimilarity is almost linearly proportional to the difference in the size of the parent subtree and the subtree rooted by c.

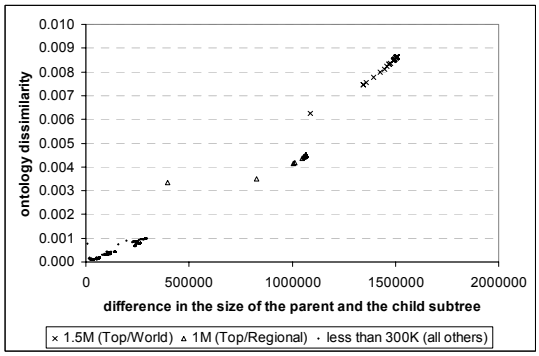

Fig. 2. Evaluation of ontologies where a concept c has been swapped with its parent. These charts explore the connection between dissimilarity and the number of instances in c's own subtree. Each choice of c is represented by one symbol (whose shape depends on the number of instances in the subtree rooted by c's parent). The x-coordinate is the difference in the number of instances between the parent's and c's own subtree.

5.3 Reassignment of instances to concepts

In the dmoz ontology, each instance is really a short natural-language document consisting of a web page title and description (usually 10–20 words). In this scenario, we follow the standard practice from the field of information retrieval and represent each document by a normalized TF-IDF vector. Based on these vectors, we compute the centroid of each concept, i.e. the average of all documents that belong to this concept or to any of its direct or indirect subconcepts. The cosine of the angle between a document vector and a concept centroid vector is a measure of how closely the topic of the document matches the topic of the concept (as defined by the set of all documents belonging to that concept). We then reassign each document to the category whose centroid is the most similar to the document vector. Thus, the hierarchical relation between concepts remains unchanged, but the assignment of instances to concepts may change considerably. This reassignment of instances to the nearest concepts resembles operations that might be used in an automated ontology construction or population approach (e.g. analogous to k-means clustering). We then measure the similarity of the new ontology (after the reassignment of documents to concepts) to the original one.

For reasons of scalability, the experiments in this section were not performed on the entire dmoz ontology, but only on its "Science" subtree. This consists of 11,624 concepts and 104,853 documents. We compare two reassignment strategies: "thorough reassignment" compares each document vector to the centroids of *all* concepts, while "top-down reassignment" is a greedy approach that starts with the root concept and proceeds down the tree, always moving into the subconcept whose centroid is the most similar to the document vector. When a leaf is reached, or when none of the subconcept centroids is more similar to the document vector than the current concept's centroid, the procedure stops and assigns the document to the current concept. This is much faster than thorough reassignment, but it has the risk of being derailed into a less promising part of the tree due to bad choices in the upper levels.

After documents are reassigned to concepts, new centroids of the concepts may be computed (based on the new assignment of documents to concepts), and a new reassignment step performed using the new centroids. The charts on Fig. 3 show the results for up to five reassignment steps. The overlap-based definition of δ_U (see eq. (2)) was used for both charts.

The upper chart in Figure 3 shows the similarity of the ontology after each reassignment step to the original ontology. As can be expected, top-down reassignment of documents to concepts introduces much greater changes to the ontology than thorough reassignment. Most of the change occurs during the first reassignment step (which is reasonable as it would be naïve to expect a simple centroid-based nearest neighbor approach using 10–20 word descriptions to accurately match the classification of the human editors working for dmoz). In fact, it turns out that 93% of documents are moved to a different concept during the first top-down reassignment step (or 66% during the first thorough reassignment step). However, the similarity measure between the new ontology and the original one is nevertheless fairly high (around 0.74). The reasons for this are: firstly, only the assignment of documents to concepts has been changed, but not the hierarchical relationship between the concepts; secondly, if documents are moved to different concepts in a consistent way, δ_U may change fairly little for most pairs of documents, resulting in a high OntoRand index value; thirdly, even though 93% of documents were moved to a different concept, the new concept was often fairly close to the original one. This is shown on the lower chart of Fig. 3, where the value of δ_U was computed between the concept containing a document in the original ontology and the one containing this document after a certain number of reassignment steps; this was then averaged over all documents. As this chart shows, even though only 7% of documents remained in the same concept during the first step of top-down reassignment, the average (over all documents) δ_U between the original and the new concept is not 0.07 but much higher – approx. 0.31.

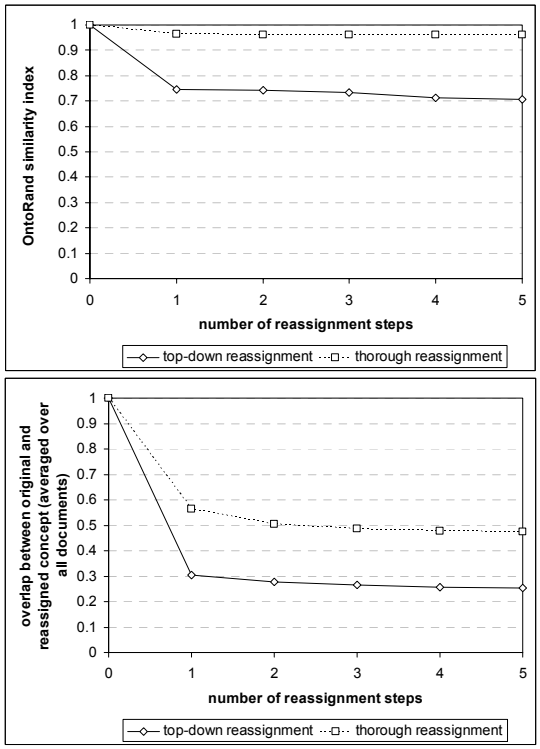

Fig. 3. Evaluation of ontology where instances have been reassigned to concepts based on their natural-language descriptions. The number of reassignment steps is used as the x-coordinate. The upper chart shows the similarity of the original ontology and the ontology after reassignment. The lower chart shows the average distance (as measured by δ_U, eq. (2)) between a concept containing an instance in the original ontology and the concept to which the instance has been reassigned.

6. Discussion and future work

The main features of our proposed approach are that it focuses on fully automated evaluation of ontologies, based on comparison with a gold standard ontology; it does not make any assumptions regarding the description or representation of instances and concepts, but assumes that both ontologies have the same set of instances. We proposed a new ontology similarity measure, OntoRand index, designed by analogy with the Rand index that is commonly used to compare partitions of a set. We propose several versions of the OntoRand index based on different underlying measures of distance between concepts in the ontology. We evaluated the approach on a large ontology based on the dmoz.org web directory. The experiments were based on several operations that modify the gold standard ontology in order to simulate possible discrepancies that may occur if a different ontology is constructed over the same problem domain (and same set of instances). The experiments show that the measure based on overlap of ancestor sets (Sec. 4.2.1) is more convenient than the measure based on tree distance (Sec. 4.2.2), because the latter requires the user to define the values of two parameters and it is not obvious how to do this in a principled way. Additionally, the tree-distance based measure is often less successful at spreading similarity values over a greater part of the [0, 1] interval; to address this issue, we propose a modified similarity measure (eq. 3'), which we will evaluate experimentally in future work. Another issue, which is shared by both similarity measures proposed here, is that the resulting OntoRand index is sometimes insufficiently sensitive to differences that occur in the upper levels of the ontology (Sec. 5.2). Sec. 5.3 indicates another possible drawback of this approach, namely that keeping the structure of the concept hierarchy and modifying only the assignment of instances to concepts may not affect the similarity measure as much as a human observer might expect.

From a purely algorithmic point of view, it would be interesting to explore if the ontology similarity measure as currently defined in Section 4.2 can be accurately computed in sub-quadratic time (in terms of the number of instances).

The experimental evaluation in Section 5 could be extended with various other operations. For example, we could split existing leaf concepts into subconcepts, either randomly or using some clustering technique. This is the converse of the operation of removing the leaf concepts described in Section 5.1. Another possibly interesting operation would be to merge two or more sibling concepts.

As the experiments with switching a concept and its parent showed (Sec. 5.2), a rearrangement of concepts in the upper levels of the tree (in our case we were switching a third-level concept and its parent, which is a second-level concept) might have only a very small effect on the similarity measure. Depending on the intended application, this may be undesirable from a human point of view because changes in the upper levels correspond to significantly different decisions regarding the conceptualization of the main concepts (especially the more abstract ones) of the domain of interest. These are important decisions that occur in the

early stages of ontology construction; therefore, it might be helpful if our similarity measures could be extended to be more sensitive to such differences in the organization of concepts in the upper levels of the ontology.

The proposed approach presented in Section 3 assumes that we are comparing two ontologies based on the same set of instances (but with different sets of concepts, different assignment of instances to concepts and different arrangement of concepts into a hierarchy). One way to extend this approach would be to allow for comparison of ontologies based on different sets of instances. In this case it is no longer possible to take a pair of instances and observe where they are placed in one ontology and where in the other, because each ontology has its own separate set of instances.

Our current approach also completely disregards concept labels, but in many practical situations these labels are an important part of the ontology and contain a lot of knowledge about the problem domain. Thus, it would be interesting to extend our approach to take concept labels or other attributes into account, e.g. via the string edit distance.

Another interesting topic for further work would be trying to evaluate an ontology "by itself" rather than comparing it to a gold standard. This type of evaluation would be useful in many contexts where a gold standard ontology is not available. One possibility is to have a partial gold standard, such as a list of important concepts but not a hierarchy; evaluation could then be based on precision and recall. Another scenario is if a gold standard is not available for our domain of interest but for some other domain, we can use that domain and its gold standard to evaluate/compare different ontology learning algorithms and/or tune their parameters, then use the resulting settings on the actual domain of our interest in the hope that the result will be a reasonable ontology, even though we do not have a gold standard to compare it to.

Acknowledgements

This work was supported by the Slovenian Research Agency and the IST Programme of the European Community under SEKT Semantically Enabled Knowledge Technologies (IST-1-506826-IP) and PASCAL Network of Excellence (IST-2002-506778). This publication only reflects the authors' views.

Bibliography and References

1. BLOEHDORN, S., HAASE, P., SURE, Y, VOELKER, J., BEVK, M., BONTCHEVA, K., ROBERTS, I., Report on the integration of ML, HLT and OM. SEKT Deliverable D.6.6.1, July 2005.
2. BRANK, J., MLADENIĆ, D., GROBELNIK, M., Automatic Evaluation of Ontologies. In: Kao, A., Poteet, S. (*eds.*), Text Mining and Natural Language Processing, Springer, 2006 (to appear).
3. BREWSTER, C., ALANI, H., DASMAHAPATRA, S., WILKS, Y., Data driven ontology evaluation. Proceedings of Int. Conf. on Language Resources and Evaluation, Lisbon, Portugal, 26–28 May 2004.
4. BURTON-JONES, A., STOREY, V. C., SUGUMARAN, V., AHLUWALIA, P., A semiotic metrics suite for assessing the quality of ontologies. Accepted by *Data and Knowledge Engineering* (2004).
5. CHAWATHE, S. S., RAJARAMAN, A., GARCIA-MOLINA, H., WIDOM, J., Change Detection in Hierarchically Structured Information. *Proc. of the ACM SIGMOD Conference*, pp. 493–504, 1996.
6. DING, L., FININ, T., JOSHI, A., PAN, R., COST, R. S., PENG, Y., REDDIVARI, P., DOSHI, V., SACHS, J., Swoogle: A search and metadata engine for the semantic web. *Proc. 13th ACM Conference on Information and Knowledge Management*, pp. 652–659 (2004).
7. EHRIG, M., HAASE, P., HEFKE, M., STOJANOVIC, N., Similarity for ontologies — a comprehensive framework. *Proc. 13th European Conference on Information Systems*, May 2005.
8. FOX, M. S., BARBUCEANU, M., GRUNINGER, M., LIN, J., An organization ontology for enterprise modelling. In: M. Prietula et al. (eds.), *Simulating organizations: Computational models of institutions and groups*, AAAI/MIT Press, 1998, pp. 131–152.
9. GÓMEZ-PÉREZ, A. Some ideas and examples to evaluate ontologies. Knowl. Sys. Lab., Stanford Univ., 1994.
10. GÓMEZ-PÉREZ, A. Towards a framework to verify knowledge sharing technology. *Expert Systems with Applications*, 11(4):519–529 (1996).
11. GROBELNIK, M, MLADENIC, D., Automated Knowledge Discovery in Advanced Knowledge Management, Journal of Knowledge Management, Volume 9, Issue 5, pp. 132-149, 2005.
12. GUARINO, N., WELTY, C., Evaluating ontological decisions with OntoClean. *Comm. of the ACM*, 45(2):61–65, February 2002.
13. HARTMANN, J., SPYNS, P., GIBOIN, A., MAYNARD, D., CUEL, R., SUÁREZ-FIGUEROA, M. C., SURE, Y., Methods for ontology evaluation. KnowledgeWeb (EU-IST Network of Excellence IST-2004-507482 KWEB), Deliverable D1.2.3, January 2005.
14. HAUSSLER, D., Convolution kernels on discrete structures. Technical report, Department of Computer Science, University of California at Santa Cruz, 1999.
15. LOZANO-TELLO, A., GÓMEZ-PÉREZ, A., Ontometric: A method to choose the appropriate ontology. *Journal of Database Management*, 15(2):1–18 (2004).
16. MAEDCHE, A., STAAB, S., Measuring similarity between ontologies. *Proc. 13th CIKM* (2002). LNAI vol. 2473.
17. MEILA, M., Comparing clusterings by the variation of information. *Proc. 16th Ann. CoLT*, 2003.
18. MEILA, M., Comparing clusterings — an axiomatic view. *Proc. ICML*, 2005.
19. MLADENIC, D., Machine Learning on non-homogeneous, distributed text data. Ph.D. thesis, University of Ljubljana, 1998.
20. MLADENIC, D., GROBELNIK, M., Feature selection on hierarchy of web documents. J. of Decision support systems, 35, 2003, 45-87.
21. PATEL, C., SUPEKAR, K., LEE, Y., PARK, E. K., OntoKhoj: a semantic web portal for ontology searching, ranking and classification. *5th ACM Workshop Web Inf. & Data Mgmt*, New Orleans, USA, 2004.
22. PORZEL, R., MALAKA, R., A task-based approach for ontology evaluation. *Proc. ECAI 2004 Workshop on Ontology Learning and Population*, pp. 9–16.
23. RADA, R., MILI, H., BICKNELL, E., BLETTNER, M., Development and application of a metric on semantic nets. IEEE Trans. on Systems, Man, and Cybernetics, 19(1):17–30 (1989).
24. RAND, W. M., Objective criteria for the evaluation of clustering methods. *J. of the American Stat. Association*, 66:846–850 (1971).
25. SPYNS, P., EvaLexon: Assessing triples mined from texts. Tech. Rpt. 09, STAR Lab, Brussels, 2005.
26. SUPEKAR, K. A peer-review approach for ontology evaluation. *Proc. Int. Protégé Conf.*, Madrid, Spain, 2005.
27. VELARDI, P., NAVIGLI, R., CUCCHIARELLI, A., NERI, F., Evaluation of OntoLearn, a methodology for automatic learning of domain ontologies. In: P. Buitelaar, P. Cimiano, B. Magnini (eds.), *Ontology Learning from Text: Methods, Evaluation and Applications*, IOS Press, 2005.
28. VÖLKER, J., VRANDECIC, D., SURE, Y., Automatic evaluation of ontologies (AEON). *Proceedings of the 4th International Semantic Web Conference*, 2005.