Evaluation Strategies for Learning Algorithms of Hierarchies

Korinna Bade[1] and Dominik Benz[2]

[1] Faculty of Computer Science, Otto-von-Guericke-University Magdeburg, D-39106 Magdeburg, Germany, Email: korinna.bade@ovgu.de
[2] Department of Electrical Engineering/Computer Science, University of Kassel, D-34121 Kassel, Germany, Email: benz@cs.uni-kassel.de

Summary. Several learning tasks comprise hierarchies. Comparison with a "gold-standard" is often performed to evaluate the quality of a learned hierarchy. We assembled various similarity metrics that have been proposed in different disciplines and compared them in a unified interdisciplinary framework for hierarchical evaluation which is based on the distinction of three fundamental dimensions. Identifying deficiencies for measuring structural similarity, we suggest three new measures for this purpose, either extending existing ones or based on new ideas. Experiments with an artificial dataset were performed to compare the different measures. As shown by our results, the measures vary greatly in their properties.

Key words: gold-standard evaluation, hierarchy, clustering, ontology learning

1 Introduction

An important task of organizing information is to induce a hierarchical structure among a set of information items or to assign information resources to a predefined hierarchy. In order to assess the quality of a learned hierarchical scheme, often an external "gold-standard" is invoked for comparison. For this task, various similarity metrics have been proposed, mostly depending on the characteristics of the applied learning procedure. This work aims at bringing together the different disciplines by presenting and comparing existing gold-standard based evaluation methods for learning algorithms that generate hierarchies. Our goal is explicitly not to identify the "best" method or metric, but to inform the choice of an appropriate measure in a given context. In the following, we will start with giving a definition of a hierarchy which abstracts from the considered learning tasks. In Section 2, we review existing evaluation strategies from the literature. Based thereon, we present an interdisciplinary framework for evaluation in Section 3, emphasizing the strong similarities of evaluation in different disciplines. The section is completed with a set of new

measures that addresses deficiencies in the currently available methods. The paper is concluded with several experiments in Section 4.

As mentioned before, different learning tasks work on different kinds of hierarchies. To make the commonalities of these structures explicit and therefore enable comparison, we define a hierarchy as follows (also compare [11]):

Definition 1. *A* hierarchy *is a triple,* $\mathcal{H} = (N, \prec, root)$ *whereby* N *is a non-empty set of nodes and* $\prec \subseteq N \times N$ *is a strict partial ordering defined on the set of nodes* N*.* $n_1 \prec n_2$ *implies that the node* n_2 *is a direct parent of* n_1 *in the hierarchy. Furthermore,* $<_{\mathcal{H}}$ *denotes the ancestor relation:* $n_1 <_{\mathcal{H}} n_2 \Leftrightarrow \exists n_{i_1}, \cdots, n_{i_r} : n_1 = n_{i_1} \prec \cdots \prec n_{i_r} = n_2$*.* $root \in N$ *is the root node of the hierarchy (*$\forall n \in N \setminus \{root\} : n <_{\mathcal{H}} root$*).*

This definition describes hierarchies as directed acyclic graphs (DAG) or poly-hierarchies with exactly one root node. A tree or mono-hierarchy is a special DAG, in which every node (except the root node) has exactly one parent: $\forall n \in N, n \neq root : |\{n' \in N | n \prec n'\}| = 1$. As finding appropriate labels for these nodes is a challenging subtask on its own (e.g., cluster labeling or concept naming), we will assume that a *lexicon* is assigned to a hierarchy:

Definition 2. *A* lexicon *assigned to a hierarchy* $\mathcal{H}$ *is a pair* $\mathcal{L}_{\mathcal{H}} = (L, F)$ *whereby* L *is a set of lexical entries intended to describe nodes and* $F \subseteq L \times N \times (0, 1]$ *is the labeling relation (*$(l, n, d) \in F$ *means that* l *is a label of* n *with* d *being the descriptiveness of* l *for* n*.).*

The descriptiveness of a label can represent, e.g., the confidence of a learning algorithm having found several possible labels for a node. If an algorithm always assigns a single lexical entry to a node, we call this a *strict lexicon*. Furthermore, learning tasks often require to assign data instances to the learned hierarchy. This *instance assignment* is separately defined as follows:

Definition 3. *An* instance assignment *to a hierarchy* $\mathcal{H}$ *is a pair* $\mathcal{I}_{\mathcal{H}} = (I, A)$ *whereby* I *is a set of instances and* $A \subseteq I \times N \times (0, 1]$ *is the assignment relation (*$(i, n, a) \in A$ *means that the item* i *is assigned to node* n *with the association strength* a*).*

Methods can either allow to assign an instance to more than one node (e.g., multi-label classification, soft clustering) or only to a single node (e.g., single-label classification, hard clustering). We will refer to the latter as *strict instance assignment*.

Several *learning tasks* can include hierarchies. Here, we consider hierarchical classification, hierarchical clustering, and ontology learning. In *hierarchical classification*, the classes in which items are classified are hierarchically related. This means the nodes in the hierarchy correspond to classes. The hierarchy $\mathcal{H}$ itself is given in advance and, therefore, is fixed. Unknown is the assignment of new instances to the hierarchy. To stick to the previous definition, $\mathcal{I}_{\mathcal{H}}$ is to be learned and evaluated.

In *hierarchical clustering*, a hierarchy of clusters is extracted from a dataset. The hierarchy itself is therefore learned by the algorithm at the same time as items are assigned to it. Both aspects (i.e., $\mathcal{H}$ and $\mathcal{I}_\mathcal{H}$) need to be evaluated. Furthermore, semi-supervised learning is a hybrid of these two distinct tasks, where part of the hierarchy is known in advance but can be further extended [3]. *Cluster labeling* aims at making an extracted cluster structure more useful by naming clusters. This task builds a lexicon $\mathcal{L}_\mathcal{H}$ in addition to a learned structure which is subject to evaluation.

Ontology learning from text is concerned with learning tasks on different levels like term extraction, concept extraction or relation learning [8]. As concept hierarchies (defined by the taxonomic relation) are a core component of ontologies, they are targeted by many ontology learning methods. Hereby, it is important to evaluate the learned hierarchy $\mathcal{H}$ as well as the concept labels $\mathcal{L}_\mathcal{H}$. Depending on the chosen method, the learned concept hierarchy is populated with instances. In such case, this instance assignment $\mathcal{I}_\mathcal{H}$ is also subject to evaluation.

2 Evaluation Strategies in the Literature

A very common approach for evaluating learning algorithms is to use a dataset as gold-standard. The learned output is compared to this gold-standard whereby a perfect match is best. To be able to do this, the dataset must include the desired learning result, i.e., a hierarchy. Its major drawback is that it usually cannot take into account that there is more than one correct solution. Nevertheless, it is often used and also the focus of this work. Other evaluation strategies include the definition of quality metrics on the result alone. These are intrinsic values like number of nodes in the hierarchy or intra-cluster similarity. Furthermore, a learning result could also be evaluated by human assessment. However, this method is very time-consuming and furthermore affected by the evaluators' subjectivity. In the following, we present an overview of gold-standard based evaluation approaches from the literature. Please note that they are all very focused to a specific problem. To the best of our knowledge, no previous work exists that tries to unify evaluation on hierarchies over several tasks.

Several work was published for *hierarchical classification*. However, only some of these consider the hierarchy in their evaluation. An easy procedure is to apply standard measures from flat classification on different subsets of classes. E.g., in [7], precision and recall was computed on different hierarchy levels and the error rate was split to distinguish between specialization and generalization error. Although these methods provide an idea of the behavior of an algorithm, it is often difficult to determine the better of two algorithms. This is avoided by giving a single measure. The authors of [14] extended precision and recall to allow for different severeness of a misclassification. They proposed two different measures, one based on category cosine similarity and

one based on category hierarchy distance. The authors of [6] defined a taxonomy based loss motivated by a document filtering setting. They associated different costs to a false classification depending on the structural relation between predicted and true class. These and other approaches to evaluate hierarchical classification usually differ mainly in their strategies of defining a "partially correct" classification. In [2], we generalized the standard measures precision, recall, and accuracy to integrate any strategy through the use of an utility function. Exemplary, this work used a strategy, which was oriented on user interaction with the hierarchy.

Comparing the structure itself often requires a *mapping* between the *node sets* as a first step. In the literature, this mapping is performed based on either the assigned lexicon or the assigned instances. In flat clustering, clusters can be mapped to gold-standard classes, e.g., by maximizing the resulting instance based evaluation measure. For hierarchical clustering, we proposed such a mapping in [3] and then applied measures from classifier evaluation. In the context of ontology learning, an example of node mapping based on concept labels (i.e., the associated lexicon) is [9]. Brank et al. [5] propose an implicit mapping based on instance assignment.

For *cluster labeling*, gold-standard evaluation is rarely used. [15] measured exact and partial label matches using precision in the top n results and mean reciprocal rank. They took into account identical terms as well as synonyms defined in the Wordnet ontology, which should be preferred over human assessment of matches. In [1], we define some measures that were inspired from evaluation in ontology learning. Both approaches compare labels on a node to node basis without integrating the hierarchy in their evaluation.

A core aspect of evaluating an *ontology learning* procedure is to compare concept hierarchies to a gold-standard. Dellschaft and Staab [9] review existing measures. They postulate a multi-dimensional approach which strictly separates the comparison on a lexical and structural level. For both levels, they adapt traditional precision and recall. While the lexical measures merely compare directly the label sets L of two hierarchies, taxonomic precision and recall integrate the structure by extracting characteristic excerpts (consisting of other nodes) for each node. These excerpts are then compared using standard precision and recall, and are finally aggregated to a global measure. The same paradigm underlies the taxonomic overlap measure proposed by Maedche [11]. Both approaches do not consider instance assignment. Brank et al. [5] complement this by proposing the OntoRand-Index, which compares two hierarchies based on how they structure a set of common instances.

On a more abstract level, several algorithms exist that measure *similarity of trees* by the tree edit distance. [4] provides a good review of existing methods. For the general case, computing the tree edit distance is computationally expensive. However, for specific cases an efficient computation might be available. Even more general are methods dealing with *graph similarity* (see [13] for a survey).

3 Interdisciplinary Comparison of Evaluation Measures

Though originating from different disciplines, the presented evaluation strategies exhibit strong similarities. In line with the tripartite representation from Section 1, we divided evaluation into three dimensions: structural, lexical, and instance assignment. The *structural dimension* is concerned with comparing the node sets and their hierarchical arrangement. A natural comparison is to require equality of both structures and penalize each deviation in a symmetric way. However, sometimes an asymmetric evaluation is more appropriate where one hierarchy is allowed to be an refinement of the other, e.g., to evaluate whether a fine grained dendrogram extracted through hierarchical clustering reflects a more coarse grained gold standard [3]. On the *lexical dimension*, two hierarchies are similar if their nodes are described using similar labels. Measures can be distinguished in terms of whether they require strict lexicons. The third dimension compares how two hierarchies *structure* a set of *instances*. Measures can support strict or non-strict instance assignment.

Theoretically, the given dimensions are independent. Consider for instance a hierarchical web directory whose category labels are translated into another language: While being equivalent on the structural and instance assignment level, the lexical similarity would decline dramatically. However, as mentioned earlier, some learning tasks touch several dimensions. E.g., in clustering, the structure is induced by a hierarchical arrangement of instances. According to Dellschaft and Staab [9], it is desirable to evaluate each dimension separately without interference between dimensions. However, if the task already connects several dimensions, a measure influenced by the same dimensions can be appropriate. Apart from this, they postulate a proportional error effect to represent correctly the degree of fatalness of an error. E.g., a missing node close to the hierarchy's root is typically judged more fatal than a missing leaf node. For this purpose, the output of the measure should use the complete available interval (often $[0; 1]$). Despite these desirable properties, the meaning of the measure should not be forgotten. Two measures might focus on the same dimension, while considering different view points on the problem.

Distinguishing existing measures in terms of our dimensions, however, gives a good starting point. This was done in the upper part of Table 1 for the measures from the literature review. The second half contains measures that we propose here to complement the given selection. As can be seen, lexical agreement and instance assignment can be measured independently of the other dimensions. However, the measures often require a mapping of the two node sets as described earlier. Structural similarity, on the other hand, can only be measured depending on either the lexicon or the instance assignment. This is necessary to identify corresponding locations in both hierarchies. It depends on the task at hand to decide which procedure is more appropriate. As can be seen, most existing work depends on the lexicon while measures based on instance assignment are rare. The Onto-Rand is the only measure in this direction, which has some restrictions on its applicability. As evalua-

Table 1. Dimensionality of different hierarchical evaluation measures (⊕ judged target dimension; ⊙ influencing dimension; ⊖ not applicable; otherwise not relevant)

dimension / measure	lexical		structural		inst. assignment	
	strict	non-strict	sym.	asym.	strict	non-strict
[14] Modified prec./recall					⊕	⊕
[6] Taxonomy-based loss					⊕	⊕
[2] Utility based prec./recall/acc.					⊕	⊕
[15] Label matches	⊕	⊕				
[1] tb/rb label similarity	⊕	⊕				
[9] Lexical prec./recall	⊕	⊖				
[9] Taxonomic prec./recall	⊙	⊙	⊕	⊕		
[11] Taxonomic Overlap	⊙	⊙	⊕	⊕		
[5] Onto-Rand			⊕	⊖	⊙	⊖
[4] Tree edit distance	⊙	⊙	⊕	⊕		
H (1)			⊕	⊕	⊙	⊖
ITP/ITR (4)			⊕	⊕	⊙	⊙
Extended OntoRand (5)			⊕	⊖	⊙	⊙

tion based on instance assignment is a very interesting problem (especially for hierarchical clustering were no lexicon is built), we propose three alternative measures. Please note that all these measures assume that the compared hierarchies are defined on the same set of instances. We denote the gold-standard with $\mathcal{H}_g$, $\mathcal{L}_g$, and $\mathcal{I}_g$ and the learned result with $\mathcal{H}_l$, $\mathcal{L}_l$, and $\mathcal{I}_l$.

First, we propose *H-Correlation*. Similar to the Rand-Index [12] for flat clustering, structure can be evaluated based on item assignment without requiring a node mapping. This is also true for the Onto-Rand. However, our measure differs in a support for asymmetric comparison as well as in the fundamental idea, which compares instance triples opposed to pairs. The sets of all instance triples $\tau = (i_1, i_2, i_3)$ for which i_1 and i_2 have a more specific common ancestor ca as i_1 and i_3 (i.e., $ca(i_1, i_2) <_{\mathcal{H}} ca(i_1, i_3)$) are compared. In its simplest form, H-Correlation can measure the overlap of the two triple sets. This can be extended by weighting individual triples differently. The symmetric correlation H_s is defined on the left in (1). Furthermore, an asymmetric correlation H_a ((1) right) can be of interest, if one hierarchy is allowed to be more detailed than the other. In (1), $\mathcal{H}_l$ is allowed to be more detailed, whereby $\mathcal{H}_l$ is more detailed than $\mathcal{H}_g$, if $N_l \supset N_g \wedge (\forall n_i, n_j \in N_g : n_i \prec_g n_j \rightarrow n_i <_l n_j) \wedge root_l \geq_l root_g$.

$$H_s = \frac{\sum_{\tau \in T_l \cap T_g} (w_l(\tau) + w_g(\tau))}{\sum_{\tau \in T_l} w_l(\tau) + \sum_{\tau \in T_g} w_g(\tau)} \qquad H_a = \frac{\sum_{\tau \in T_l \cap T_g} w_g(\tau)}{\sum_{\tau \in T_g} w_g(\tau)} \tag{1}$$

Here, we propose and compare the following two different triple weights, whereby w_1 gives equal weight to each triple and w_2 gives equal weight to each hierarchy node through node specific normalization:

$$w_1(\tau) = 1 \qquad w_2(\tau) = 1/|\{\tau' | ca(i_1, i_3) = ca(i'_1, i'_3)\}| \tag{2}$$

The second measure is an adapted version of taxonomic precision and recall [9]. Its original version requires a node mapping between both hierarchies, which is done by matching node labels. This mapping is replaced by an examination of instance assignment in our *instance-based taxonomic precision ITP* and *recall ITR*, following the assumption that the instances assigned to the nodes in a hierarchy do exhibit a sufficient degree of topic specificity. For each instance i, we extract a characteristic excerpt from each hierarchy, namely the *instance-based semantic cotopy* $isc(i, \mathcal{H}) = \{j \in I | (j, n, s) \in A \wedge (i, m, t) \in A \wedge (n \leq_{\mathcal{H}} m \vee m \leq_{\mathcal{H}} n)\}$. This excerpt contains all instances assigned to the same, sub- or super-nodes of the nodes i is assigned to. Two excerpts of an instance are compared as in (3), whereby *itp* measures local precision and *itr* local recall. The global precision and recall values then combine the individual local values as shown in (4) for the precision.

$$itp(i, \mathcal{H}_l, \mathcal{H}_g) = \frac{|isc(i, \mathcal{H}_l) \cap isc(i, \mathcal{H}_g)|}{|isc(i, \mathcal{H}_l)|} = itr(i, \mathcal{H}_g, \mathcal{H}_l) \tag{3}$$

$$ITP(\mathcal{H}_l, \mathcal{H}_g) = \frac{1}{|I|} \sum_{i \in I} itp(i, \mathcal{H}_l, \mathcal{H}_g) \tag{4}$$

The third measure *extends* the *OntoRand Index* to non-strict instance assignment. In its original version, the basic idea is to represent a hierarchy as a vector. The similarity of two hierarchies is then the similarity between the two vectors. Every vector dimension corresponds to a pair of instances i_1, i_2. The entry at the respective dimension contains a distance measure $\delta(n_1, n_2)$ between the nodes n_1 and n_2 to which i_1 and i_2 are assigned. This requires a strict instance assignment. To extend the measure to the non-strict case, the δ-function is redefined for node sets instead of single nodes as follows:

$$\dot{\delta}(N_1, N_2) = \sum_{n_1 \in N_1, n_2 \in N_2} \varpi_{1,2} \delta(n_1, n_2) \tag{5}$$

The weighting factor $\varpi_{1,2}$ can be used to weight certain node combinations differently, e.g., by their instance association strength. In this work, we assume equally distributed association strength, which corresponds to averaging over all possible combinations: $\varpi_{1,2} = \frac{1}{|N_1||N_2|}$. Summarizing, the Extended OntoRand Index (EOR) is identical to the original version in [5] except that the δ-function is replaced by our $\dot{\delta}$.

4 Experiments and Conclusion

We performed several experiments to compare the proposed measures in terms of their behavior to different types of structural differences. For comparability,

we picked a scenario, for which all three measure are applicable, i.e., a symmetric comparison with strict instance assignment. However, be reminded that the applicability of the different measures varies and, depending on the task, not all measures might be available. In order to avoid any kind of bias introduced by a real-world hierarchy, we chose to create an artificial dataset as gold standard. It consisted of 190 nodes with 50 instances assigned to each node (except root, making it a total of 9450 instances). The branching factor varied between 2 or 3, the depth was 5.

Starting from our gold standard, we created six different experimental setups by systematically inducing different structural errors. We then measured the similarity between the original and the modified hierarchy with each of our proposed measures. The results of these experiments are shown in Figure 1. In setting (a), the hierarchy was cut at a certain depth level. Instances assigned to nodes that were removed were assigned to the closest ancestor node that remained in the hierarchy. This scenario was already used by Brank et al. [5] for their evaluation of the OntoRand. It simulates a top-down learning algorithms that stops its refinement too early, e.g., caused by an erroneous stopping criterion. In setting (b), the hierarchy was changed to binary by introducing an empty intermediated node combining two of three child nodes in the cases of branching factor 3. The diagram varies between no change and replacing all 40 3-child-nodes. This setting might occur, e.g., as artifact of a clustering procedure. In (c) and (d), nodes were removed either on level 3 or 5. Their instances and child nodes were assigned to their parent node. Both diagrams are in the same range, which equal the maximum number of nodes on level 3. In (e) and (f), instances/nodes (with their instances but without their child nodes) were moved randomly to a different place in the hierarchy. The diagrams range up to about 50% of all instances/nodes moved.

Setting (a) clearly shows different sensitivity of the algorithms towards the hierarchy levels. H_2 is least affected by the hierarchy level and therefore has the largest drop in similarity as lower levels have more nodes. The other extreme is H_1, which hardly looses similarity as long as the top level separation is not violated. The other measures lay in between. Regarding ITP and ITR, only ITP reacts in this case. ITR would cover the opposite case where the hierarchy would be split further than required. Error type (b) cannot be measured with ITP/ITR at all as they ignore the empty nodes. The peaks in the H_1 and EOR curve are an artifact of the level sensitivity. A closer look at the data revealed that they occur, when nodes were inserted at high levels. Setups (c) and (d) further show this sensitivity. H_2 has an identical curve on both levels. The other measures react stronger on the higher level. It holds: The less sensitive a measure is for depth (except not sensitive at all), the stronger this difference can be seen. In condition (e), all measures show a similar linear decrease, except EOR, which seems to be less influenced by pure instance movement. An interesting observation in (f) is the fast decrease of H_1, which showed much less reaction in the other conditions. We attribute this to the fact that nodes were often move to completely different parts of the hierarchy,

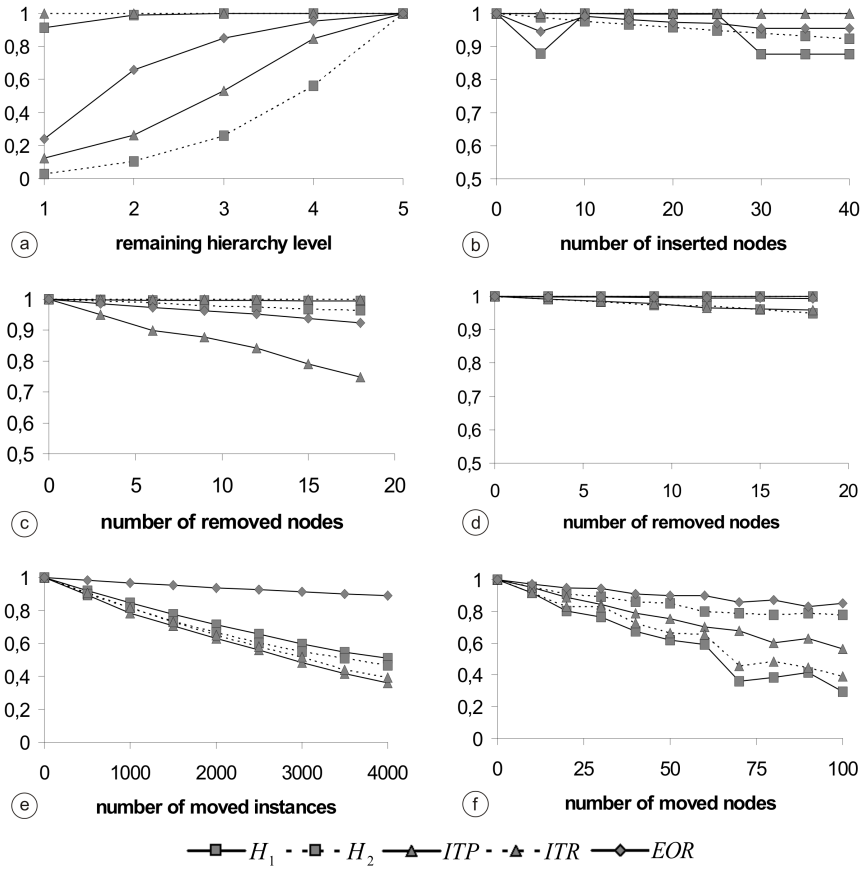

Fig. 1. Experimental Results: a) hierarchy cutting, b) inserting empty intermediate nodes, c/d) removing nodes on level 3/5, e/f) moving instances/nodes

therefore introducing a significant error. Furthermore, it is interesting that EOR, which also has a quite large sensitivity to hierarchy level reacts the fewest. This shows that both measures have a fundamentally different concept of measuring similarity. Summing up, we want to rank our measures according to increasing level sensitivity: H_2, EOR, ITP/ITR, H_1. However, this is not a sufficient distinction as we showed highly different behavior. There is no clearly best measure and it is an individual decision of which measure best reflects the evaluation purpose. An important question to assess is: How many specific errors count as much as a single general error? This question is not easy to answer. However, increasing sensitivity leads to less smooth behavior of the measure. This could be an argument against (too strong) level sensitivity.

Concluding the paper, we want to point out that the contribution of this paper is twofold: First, we assembled an interdisciplinary pool of evaluation

methods for learning algorithms of hierarchies, embedded in a generic framework. Second, we proposed new instance-based measures for structural hierarchy comparison and analyzed their properties experimentally. Our experiments describe the characteristics of the measures and can be used as basis for an individual decision according to the specific evaluation need. The most obvious difference can be found in the sensitivity to hierarchy levels. Using several measures should provide clear evidence how and where two hierarchies differ. Our methodology and results lay the groundwork for further work in this direction, which we are also pursuing. Furthermore it should be noted that all measures suffer from the same problem as the originally proposed Rand index. An adjustment as proposed for the Rand index in [10] is necessary to allow comparison of results over different gold standards.

References

1. K. Bade and M. Hermkes. Collection browsing through automatic hierarchical tagging. In *Proceedings of the 5th International Conference on Adaptive Hypermedia and Adaptive Web-based Systems*, 2008.
2. K. Bade, E. Hüllermeier, and A. Nürnberger. Hierarchical classification by expected utility maximization. In *Proceedings of the 2006 IEEE International Conference on Data Mining*, 2006.
3. K. Bade and A. Nürnberger. Creating a cluster hierarchy under constraints of a partially known hierarchy. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pages 13–24, 2008.
4. P. Bille. A survey on tree edit distance and related problems. *Theoretical Computer Science*, 337(1-3):217–239, 2005.
5. J. Brank, D. Madenic, and M. Groblenik. Gold standard based ontology evaluation using instance assignment. In *Proc. of 4th EON Workshop*, May 2006.
6. L. Cai and T. Hofmann. Hierarchical document categorization with support vector machines. In *Proc. of the 13th ACM CIKM*, pages 78–87, 2004.
7. M. Ceci and D. Malerba. Hierarchical classification of html documents with webclassii. In *Proc. of the 25th European Conference on IR*, pages 57–72, 2003.
8. P. Cimiano. *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Springer, Nov. 2006.
9. K. Dellschaft and S. Staab. On how to perform a gold standard based evaluation of ontology learning. In *Proc. of ISWC-2006 Intern. Semantic Web Conf.*, 2006.
10. L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, December 1985.
11. A. Maedche. *Ontology Learning for the Semantic Web*. Kluwer Academic Publishing, Boston, 2002.
12. W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:622–626, 1971.
13. A. Schenker, H. Bunke, M. Last, and A. Kandel. *Graph-Theoretic Techniques for Web Content Mining*. World Scientific Publishing Co., Inc., 2005.
14. A. Sun and E. Lim. Hierarchical text classification and evaluation. In *Proc. of the 2001 IEEE International Conference on Data Mining*, pages 521–528, 2001.
15. P. Treeratpituk and J. Callan. Automatically labeling hierarchical clusters. In *Proc. of the Int. Conf. on Digital Government Research*, pages 167–176, 2006.