Quality of Subsumption Hierarchies in Ontologies

Geir Solskinnsbakk[1], Jon Atle Gulla[1], Veronika Haderlein[2], Per Myrseth[2], and Olga Cerrato[2]

[1] Department of Computer and Information Science
Norwegian University of Science and Technology
Trondheim, Norway
{geirsols, jag}@idi.ntnu.no
[2] Det Norske Veritas (DNV)
Oslo, Norway
{Veronika.Haderlein, Per.Myrseth, Olga.Cerrato}@dnv.com

Abstract. Ontologies are becoming increasingly more popular tools for many tasks, such as information integration, information retrieval, knowledge management and extraction etc. The cost and complexity of developing good ontologies is high, and therefore it is important to be able to verify the ontology and detect flaws early. In this paper we propose an approach to expose desirable properties of ontological structures. The approach is based on an ontological profile which is an ontology extended with a vector of weighted terms describing the semantics of each concept of the ontology. We describe four hypotheses for the relations among the classes of the ontology and perform experiments to verify them. Our initial findings are that the experiments support the hypotheses.

1 Introduction

With the emergence of the Semantic Web [2] and Semantic Web related technologies, ontologies are becoming steadily more important. This also means that the quality aspect of ontologies is becoming even more important. Ontologies are formal specifications of shared conceptualizations [3] and are used for a wide range of tasks, such as information integration, information retrieval, knowledge management and extraction etc. Designing ontologies is an expensive and time consuming task, and adding to the complexity is the problem of modelers not being appropriately familiar with the domain, requiring domain experts which are not expert modelers to chip in. On the other hand, there has been quite a lot of work done on automatic ontology building. However, the task of ontology development is complex, and automatic ontology building is a hard task. Therefore it is important to be able to verify the consistency of the ontology and find mistakes as early as possible. We thus present in this paper a method for exposing desirable properties in the hierarchical structure of the ontology with respect to subsumption. The proposed approach is based on the concept of ontological profile, which is an ontology extended with term vectors for each

concept. The term vector describes the semantics of the concept with respect to an underlying text collection. The objective of our approach is to be able to specify and verify desirable properties of ontological structures. Our initial experiments show encouraging results towards the objective. The rest of the paper is structured as follows; Section 2 gives a short introduction to some of the most important related work on the field of ontology taxonomy verification, while Section 3 gives a short introduction to the concept of ontological profiles. Section 4 gives an overview of the approach, and specifies hypotheses about structure properties. Details about the experiment are found in Section 5, while the results are reported and commented in Section 6. Finally the paper is concluded in Section 7.

2 Related Work

Guarino et al. presents in [4] the methodology OntoClean for validating taxonomic relations in ontologies. The methodology is based on philosophical conceptions like essence, identity, and unity. These conceptions are used as metaproperties on the taxonomic structure describing the classes ([4] uses the term property about classes). The metaproperties described by the authors are Rigidity (R), Unity (U), Dependence (D), and Identity (I). The taxonomic structure of the ontology is verified by first tagging the concepts with metaproperties, and next the tags in the subsumption hierarchy are analyzed. The methodology imposes constraints on the subsumption of concepts with certain combinations of metaproperties. By removing subsumption relations that violate these constraints, relationships that were ill-defined from the designers part are avoided. Highly related to the work of Guarino et al. is the work of Völker et al. [9], which describes a tool, AEON, to automatically tag a RDF/OWL ontology with metaproperties. The authors use positive and negative evidence found in a large corpus (the Web) as basis for tagging classes with metaproperties. For each metaproperty the authors use a set of patterns to gather positive and negative evidence by querying the Google API. The patterns are in the form of a natural language string with a variable that is replaced by the concept in question. One example of such a pattern is *"is no longer (a|an)? x"*[9]. By replacing the variable x with the concept in question (e.g. "student") this specific pattern provides negative evidence for the Rigid metaproperty (a class is considered Rigid when an instance can not stop being an instance of that class, e.g. a human can not stop being a human, on the other hand, a student can stop being a student). The results from Google are analyzed by part-of-speech tagging the result to remove results that give false match for the pattern. Finally they use a classifier to decide whether or not a metaproperty is applicable to the concept based on the evidence.

3 Ontological Profiles

An ontological profile is an ontology extended with vectors of terms for each concept of the ontology. These terms are regarded as semantic descriptions of the concept at hand, based on an underlying document collection. Moreover, the terms are weighted to reflect the strength of the relation between the concept and the term. By using a document collection as the basis for the ontological profile, both the terms and their weights are reflections of the real world usage of the ontology concepts. For a definition of concept vector see Definition 1.

Definition 1. *Concept Vector. The definition given here is adapted from Su[8]. Let T be the set of n terms in the document collection used for construction of the ontological profile. $t_i \in T$ denotes term i in the set of terms. Then the concept vector for concept j is defined as the vector $C_j = [w_1, w_2, \ldots, w_n]$ where each w_i denotes the semantic relatedness weight for each term t_i with respect to concept C_j.*

The method of constructing an ontological profile is systematic, in that we have defined a way of assigning documents to and creating vectors from these documents to represent concepts of the ontology. For a more detailed description of ontological profiles, see [7].

We have done (and are still doing) research on using ontological profiles for information retrieval (IR) purposes, for an introduction to the use of ontological profiles for IR see for example [7]. Further, ontological profiles have been used for ontology alignment purposes [8].

4 Approach

Our approach of using vectors of weighted terms associated with ontological concepts, gives us a sort of semantic description of the notions used in real world texts describing the concept. The terms contained in the vectors are restricted to stemmed versions of certain word classes (part-of-speech tagging), with certain frequent words removed (details described in Section 5). The main objective of our approach is to compare these text based descriptions of concepts, concept vectors, with each other. The comparison is used as a basis to define and verify desirable properties in the subsumption hierarchy of the ontology.

To give the reader a real example of the vectors created from the DNV ontology, we can take a look at the concept *careers* which has the following *phrase vector* definition (only top 5 terms are shown with weights in subscript):

$$C_{careers} = [\text{“high ambitions”}_{5.9}, \text{“extra dimension”}_{5.9}, \text{“profiling film”}_{5.9}, \text{“vacant positions”}_{5.2}, \text{“dnv uk”}_{3.3}]$$

As we see from the vector these terms are quite reasonable for what you would expect from a recruitment page. Another example from the ontology is

the concept *maritime* which is an import business area for DNV. The *phrase vector* is represented as follows:

$$C_{maritime} = [\text{“ship classification”}_{3.5},\ \text{“maritime industry”}_{3.1},\ \text{“strong base”}_{2.9},\ \text{“printed editions”}_{2.9},\ \text{“ships life”}_{2.9}].$$

Since the vector representation carries a semantic description of the concept, we should be able to find interesting properties that should hold for hierarchical relations of good quality by comparing the vectors of super classes and sub classes. We use two different approaches for the comparison of the vectors. The first is cosine similarity [1], and the second is reducing the vectors to sets (by disregarding the weights of the vectors) and performing set operations on the resulting sets. The cosine similarity is calculated as in Equation 1, where C_i and C_j are the concept vectors for concept i and j respectively, $w_{n,l}$ is the weight for term n in concept vector l, t the total number of terms, and $sim(C_i, C_j)$ is the cosine similarity between C_i and C_j.

$$sim(C_i, C_j) = \frac{\sum_{n=1}^{t} w_{n,i} \times w_{n,j}}{\sqrt{\sum_{n=1}^{t} w_{n,i}^2} \times \sqrt{\sum_{n=1}^{t} w_{n,j}^2}} \tag{1}$$

The approach does not consider the semantics of separate words other than removing words that are stop words, and words that are not tagged as nouns using part-of-speech-tagging. There is one exception to the last point, we include adjectives that are parts of noun phrases (described in Section 5). Moreover, we do not consider the implicit relations between words, only how the words sum up to describe the concepts of the ontology.

We have proposed four different hypotheses about properties of the hierarchical relations in the ontologies based on the two measures just described (cosine similarity and set relations).

4.1 Hypotheses

We will in this subsection describe our four hypotheses and argue for why they seem reasonable. For the sake of making the explanation of the hypothesis simpler, we will use the notation described below. A concept C of the ontology has n sub classes, C_i. All concepts C_i (under C) are then said to be siblings at the same abstraction level of the ontology. Further, the super concept's vector representation is given by S, and its corresponding set representation (by disregarding the weights of S) is given by $S^{'}$. Likewise, for the sub classes, the vector representation of sub class C_i is given by U_i and the corresponding set representation is given by $U_i^{'}$.

Hypothesis 1. *The relationship between super and sub class is stronger than between the sub classes.*

In other words, Hypothesis 1 states that we expect to find that the relationship between a class and its subsumer is stronger than the relationship between the class and its siblings on the same level of the hierarchy. Put in terms of the ontology relations, evidence supporting this hypothesis should be for each S and all its corresponding sub vectors U_i, U_j: $sim(S, U_i) > sim(U_i, U_j)$ and $sim(S, U_j) > sim(U_i, U_j)$. The argument for this is that the sub-super relation carries some commonalities that we should be able to observe. On the other hand the commonalities of the sibling relations are mainly carried by the relationship with the super class relation, and should thus not be as prominent.

Hypothesis 2. *Characterizations of super class and sub class overlap semantically, but refer to different levels of abstraction*

Hypothesis 2 expresses that although there is a relation between the super class and its sub class, the set representation of the two should be different. The motivation for this is that while the super class has a broader definition, touching some of the relevant aspects of the sub class, the sub class should have its own, narrower description, fleshing out on the more detailed aspects. Evidence supporting this hypothesis should be found in the form $S^{'} \setminus U_i^{'} \neq \emptyset$ and $U_i^{'} \setminus S^{'} \neq \emptyset$ thus signaling that neither set contains fully of the other set.

Hypothesis 3. *Commonalities among subclasses are defined by their super class.*

By Hypothesis 3 we mean that the super class, being the more general class in the hierarchy, defines some least common set that should be found amongst the sub classes. In other words, we expect that the super class has a partitioned terminology, one describing abstract features, and one describing more specific features of the concept. For this hypothesis to be true, we would expect to find a common terminology amongst the sub classes that also would be found in the super class' terminology. Specifically, we would expect to find that the specific part of the super class' description is shared with the intersection of the sub classes. In terms of set relation we can say that evidence supporting this hypothesis should be $(\bigcap_{i=1}^{n} U_i^{'}) \setminus S^{'} = \emptyset$.

Hypothesis 4. *There are abstract features of a super class that are not shared by any subclass.*

Hypothesis 4 states that the super class is defined at a higher level of abstraction using terminology that is not directly applicable to the lower level of abstraction of the sub class. Using the same argumentation as in Hypothesis 3, we can say that the terminology is partitioned. Whereas Hypothesis 3 tests the commonalities between the super class and sub classes (specific features of the super class), we are here interested in the abstract features of the super class. The super class should describe the concept at a higher level, and thus contain terminology that is not interesting to deal with on the more detailed level of the sub class. Evidence supporting this hypothesis should thus be $S^{'} \setminus (\bigcup_{i=1}^{n} U_i^{'}) \neq \emptyset$ and $|S^{'} \setminus (\bigcup_{i=1}^{n} U_i^{'})| < |S^{'}|$.

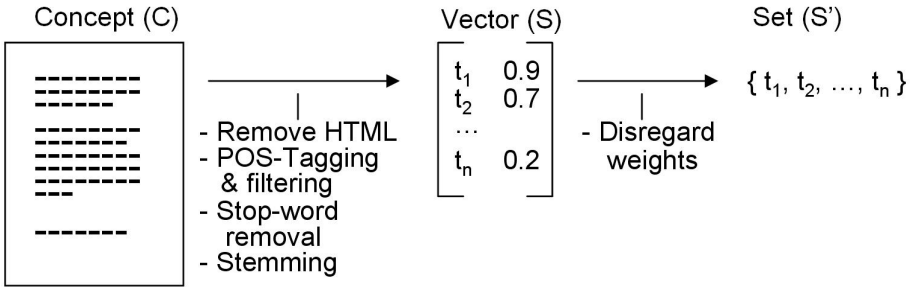

Fig. 1. Overview of vector construction.

5 Experiment

The data set we have based our experiments on is the web site of a large corporation, DNV[3], which has activities spanning globally. The web site for this company has been downloaded for two separate temporal snapshots, namely 2004 and 2008. We made a simple parser application that uses the site map of the site as a basis, all pages listed in the site map were attempted downloaded. The site map for 2004 contained 207 files of which we successfully downloaded all, while the site map for 2008 contained 424 files, which we also managed to successfully download. However, only 369 of the files from 2008 contained content (the rest were subject to "http 404 file not found" or internal server errors at the company). As the actual data source for the files we used the web site of the Internet Archive [4] and their Wayback Machine.

Further we have used this site and its hierarchy as a ontology, using the structure of the site to create a subclass hierarchy. This is not considered an actual ontology by many, but our main purpose in this experiment is to look at the hierarchical relationships between concepts, so for our use this will suffice. Furthermore, we assume that the ontology of DNV is good with respect to subsumption, as it has been developed over many years and is continuously updated. Thus we deem that the results we get from our experimental results should be valid for concluding about how a good ontology is structured.

Figure 1 shows the overall techniques used during construction of the concept vectors, discussed in more detail below. For each node in the ontology (web page) we created a concept vector. As we regard each of the web pages in the hierarchy as a single ontology concept, it is quite straight forward to assign documents to each concept. We simply view the single document found at the concept as its textual description.

Preprocessing and cleaning of the html documents is the first step for creating a concept vector for each concept in the ontology. First, we remove any

[3] http://www.dnv.com
[4] http://www.archive.org

html tags, script tags, common structures (such as menus) etc. from the documents, leaving us with clean text files. Since html tagging is mainly a layout formatting, authors are not always very good at writing grammatically correct pages with respect to the textual content, which hampers the results of the next step in the process, part-of-speech (POS) tagging . This lead us to apply the following solution. Assuming that any text within certain types of tags is either a sentence/paragraph or a stand-alone collection of words (e.g. the entry in a bullet point), we inserted extra punctuations (".") into the html before the html was removed. By not applying this solution we could in the worst case end up with a whole table as a full sentence.

For POS-tagging we used the Stanford Tagger v1.6 [5]. The POS tags were further used to remove words of unwanted word-classes, and recognize phrases in the text. As a basis for the phrase extraction we use a set of POS tag patterns slightly different than the ones suggested by Justeson and Katz [5]. The tag patterns we use are shown in Table 1, where N is a noun (we have not differentiated between words within the noun class), and J is an adjective (also here we do not differentiate between words within the adjective class).

Table 1. Part-of-speech tag patterns for phrase recognition, based on the patterns suggested in [5].

Length	Pattern
2	NN, NJ
3	NNN, NJN , JNN, JJN
4	NNNN

All phrases found by the tag patterns were added to the concept vector without any frequency filtering. This will of course result in some noise, but we found that filtering on a frequency of 2 would disqualify a large number of good phrases found in the text. Thus the benefit of adding more good phrases was considered to be higher than the disadvantage of the small amount of noise that was added. In addition to the phrases found based on the tag patterns in Table 1, we also added single nouns (NN, NNS, NNP, NNPS) that appear in more than two documents to the concept vectors.

Next the phrases were split into its sub terms, and all these were added to the concept vectors (using the frequency of the phrase). We note that in the step of breaking up the phrases, there are still some terms that can be adjectives, although we specified that only nouns (single terms) should be added to the vector. The argumentation we use is that the adjectives present in a noun phrase are more important for the semantic description of the concept than other "stand-alone" adjectives, and thus are added to the vector.

[5] http://nlp.stanford.edu/software/tagger.shtml

To make sure that the vectors are containing only the most meaning bearing terms, we next remove the stop words (if any, since the POS-filtering should have removed a substantial portion). The final processing of the terms is to apply stemming using the Porter stemming algorithm [6]. This ensures that terms having the same general meaning are collapsed to a single term. Finally, the weight of the terms are calculated based on the familiar $tf \times idf$ score[1] depicted in Equation 2, where $w_{i,j}$ is the weight of term i in concept vector j, $freq_{i,j}$ is the frequency of term i in the concept vector j, $max_l freq_{l,j}$ is the frequency of the most frequent term l in concept vector j, N is the number of concept vectors, and n_i is the number of concept vectors containing term i.

$$w_{i,j} = \frac{freq_{i,j}}{max_l freq_{l,j}} \times log \frac{N}{n_i} \tag{2}$$

To compute the set operations needed for Hypothesis 2-4, we disregard the weight information for the terms, and regard each vector as a set, adding access methods to perform set operations on the resulting set. The cosine similarity calculations were performed according to Equation 1.

6 Results

We will in this section present and comment upon the results obtained from our experiments. The experiment consisted of running two tests for each of the four hypotheses. For all the tests we located all the classes that had two or more sub classes. For the 2004 collection this left us with 26 super classes, and for the 2008 collection we were left with 60 super classes. Please refer to the example from Section 4.1 for the notation used to describe the classes/vectors/sets.

The first part of the experiment was concerning Hypothesis 1 (which deals with the strength of the sub-super relation in contrast to the sibling relations). We specified the success criteria for the hypothesis to be $sim(S, U_i) > sim(U_i, U_j)$ and $sim(S, U_j) > sim(U_i, U_j)$. We had a total of 26 (2004) and 60 (2008) classes with 2 or more sub classes, and for the sub-super relation analysis we examined the cosine similarity between the super class and each of its sub classes (totaling 172 relations for 2004 and 344 for 2008). For each of the super classes we examined the sibling relation (according to how sibling relations were defined in Section 4.1) for each pair of sub classes. The number of sibling relations analyzed totaled 744 for the 2004 collection, and for the 2008 collection 1482. Looking at the results from Table 2 we see that the mean cosine similarity between super and sub concepts is higher than the similarity between siblings. This seems to support the hypothesis, and we see that the results seem to agree for both collections.

Next, we look at the part of the experiment concerning Hypothesis 2 (super class and sub class are different). Recall that we specified the success criteria for the hypothesis to be $S^{'} \setminus U_i^{'} \neq \emptyset$ and $U_i^{'} \setminus S^{'} \neq \emptyset$. Looking at the results in Table 3 we can see that this indeed seems to be the case. We see that both result sets are quite large, meaning that there is some partial overlap between

Table 2. The results of the experiment for Hypothesis 1.

Variable	2004	2008
Mean sub-super similarity	0.347	0.348
Mean sibling similarity	0.197	0.219
Number of concepts having a mean sibling similarity larger than mean sub-super similarity	5	6

the two, and that both concepts seem to have their own specificities. The super class should have some general more abstract terms which are more appropriate at a more abstract level, while the sub class should contain some terms that are more specialized and appropriate at a more detailed level. It is however not an easy task to analyze what an optimal result for this test is. Very large result sets would indicate little overlap, while small result sets indicates a high degree of overlap. The optimal size of the result set remains an open issue.

Table 3. The results of the experiment for Hypothesis 2.

Variable	2004	2008
Mean number of terms in $S^{'}$	65.5	71.3
Mean number of terms in $U_i^{'}$	65.8	71.4
Mean number of terms in $S^{'} \setminus U_i^{'}$	43.3	44.8
Mean number of terms in $U_i^{'} \setminus S^{'}$	41.7	46.6

The third part of the experiment concerns whether commonalities among the sub classes are defined by their super class (refer to Hypothesis 3). Further we specified the evidence criterion for supporting this hypothesis as being $(\bigcap_{i=1}^{n} U_i^{'}) \setminus S^{'} = \emptyset$. From the results in Table 4 we can see that this indeed seems to be the case. For 2004 we have 18 (out of 26) empty result sets, while we have for 2008 28 (out of 60) empty result sets. Further we note that the mean size of the result sets is quite low, 0.7 and 3.7 for 2004 and 2008, respectively. The interpretation of this result is that there is some commonality between the sub classes that is also defined by the super class. The result in our opinion is quite clear, as supported by both collections

The last part of the experiment was concerning Hypothesis 4. The test run was based on subtracting the super class from the union of the sub classes $(S^{'} \setminus (\bigcup_{i=1}^{n} U_i^{'}))$, and the evidence criteria were specified as $S^{'} \setminus (\bigcup_{i=1}^{n} U_i^{'}) \neq \emptyset$ and $|S^{'} \setminus (\bigcup_{i=1}^{n} U_i^{'})| < |S^{'}|$. The results are depicted in Table 5 and we can see that for each of the collections there is a single empty result set, meaning that

Table 4. The results of the experiment for Hypothesis 3.

Variable	2004	2008
Mean number of terms in $S^{'}$	65.5	71.3
Mean number of terms in $(\bigcap_{i=1}^{n} U_i^{'})$	13.7	18.4
Mean number of terms in $(\bigcap_{i=1}^{n} U_i^{'}) \setminus S^{'}$	0.7	3.7
Empty result sets	18	28

the entire super vector is contained in the union of the sub vectors. This result shows that the super set does not contain any specific semantics not carried in the sub sets. On the other hand, for the remaining sets we see that the result set contains on average 15.3 and 23 terms for 2004 and 2008, respectively. We see that the results (both for 2004 and for 2008) agree quite well with the evidence criterion given. We interpret this as that the semantics of the super concept can be split in two; one general (abstract) part most appropriate for the higher level, and one more specific part that is also shared with the lower level (the sub classes).

Table 5. The results of the experiment for Hypothesis 4.

Variable	2004	2008
Mean number of terms in $S^{'}$	65.5	71.3
Mean number of terms in $(\bigcup_{i=1}^{n} U_i^{'})$	222.2	223.7
Mean number of terms in $S^{'} \setminus (\bigcup_{i=1}^{n} U_i^{'})$	15.3	23.0
Empty result sets	1	1

7 Conclusion

In this paper we have presented an approach for describing properties in a subsumption hierarchy of an ontology, and run tests that verify that our hypotheses seem reasonable. We have used as basis the DNV ontology defining the structure of the www.dnv.com web site. Since this ontology has been developed for several years and is subject to continuous update, we deem it as a good ontology for our evaluation. The first thing we can note from the results is that there seems to be a stronger relation between a class and its subsumer, than between a class and its siblings on the same level of abstraction. This is a nice result pointing out how the relations should be between sub/super class and siblings in the ontology. Further, we found that there is indeed a difference between the super

class and the sub class, while they still retain some similarity. We interpret this as the super class having a broader definition, and carrying a vocabulary more suited for the higher level of abstraction, while the sub class has a more detailed vocabulary geared towards the lower level of abstraction. This may point out that classes in a subsumption hierarchy that overlap to a high degree possibly should not have the subsumption relation.

We also found that the commonalities between the sub classes (represented as the intersection of the sub classes) defines a common set that also can be found in the super class. If the commonalities were not found in the super class this could possibly be a indication that the relation is inappropriate, or that we really are dealing with concepts that should be even further down the hierarchy (a missing class between the super and sub classes). Finally we found that the super class and the sub classes overlap partially semantically. The super class contains one part specifying its more abstract semantics (not shared by the sub classes), while the sub classes contain one part specifying their more detailed nature.

All of these findings are supported both in the 2004 and the 2008 version of the ontology, even though the ontology itself has evolved over the time period. This indicates that the ontology has been updated by sound principles.

There are however some weaknesses in our approach. First, we do not look at the semantic relations between the words in the sets/vectors. Second, and this is more of a concern towards the validity of our experiments, the amount of text used to construct the sets/vectors is limited. It would thus be an interesting point for further work to do a more thorough analysis of the approach with a larger data set. Lastly, our approach does not specify what the classes mean, rather how they are defined by words. Guarino [4] using the OntoClean methodology tries in contrast to define the semantics of the classes.

Acknowledgment. This research was carried out as part of the LongRec project, project no. 176818/I40, funded by the Norwegian Research Council.

References

1. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval, ACM Press, New York, 1999.
2. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American, May, 2001.
3. Gruber, T.R.: A translational approach to portable ontologies. Knowledge Acquisition 5(2), 1993.
4. Guarino, N., Welty, C. A.: An overview of OntoClean. In S. Staab, and R. Studer (Eds.): Handbook on ontologies. International Handbooks on Information Systems, Springer, 2004.
5. Justeson, J., Katz, S.: Technical terminology: some linguistic properties and. an algorithm for indentification in text. Natural Language Engineering 1(1):9-27, 1995.
6. Porter, M.F.:An algorithm for suffix stripping, Program, 14(3) pp 130-137, 1980.

7. Solskinnsbakk, G., and J. A. Gulla: Ontological Profiles as Semantic Domain Representations. In: NLDB 2008: 13th International Conference on Applications of Natural Language to Information Systems, London, UK, 24-27 June 2008.
8. Su, X.: Semantic Enrichment for Ontology Mapping. PhD Thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2004
9. Völker, J., Vrandecic, D., Sure, Y.:Automatic Evaluation of Ontologies (AEON). Proceedings of 4th International SemanticWeb Conference, ISWC 2005 Galway, Ireland. LNCS 3729, Springer, 2005.