

Ontology Evaluation and Ranking using OntoQA

Samir Tartir and I. Budak Arpinar
Large-Scale Distributed Information Systems Lab
Department of Computer Science
University of Georgia
Athens, GA 30602-7404, USA
{tartir,budak}@cs.uga.edu

Abstract

Ontologies form the cornerstone of the Semantic Web and are intended to help researchers to analyze and share knowledge, and as more ontologies are being introduced, it is difficult for users to find good ontologies related to their work. Therefore, tools for evaluating and ranking the ontologies are needed. In this paper, we present OntoQA, a tool that evaluates ontologies related to a certain set of terms and then ranks them according a set of metrics that captures different aspects of ontologies. Since there are no global criteria defining how a good ontology should be, OntoQA allows users to tune the ranking towards certain features of ontologies to suit the need of their applications. We also show the effectiveness of OntoQA in ranking ontologies by comparing its results to the ranking of other comparable approaches as well as expert users.

1. Introduction

The Semantic Web envisions making the content of the web processable by computers as well as humans [5]. This is mainly accomplished by the use of ontologies which contain terms and relationships between these terms that have been agreed upon by members of a certain domain (e.g., the Gene Ontology (GO) [30] and other ontologies in biology such as the Open Biology Ontologies (OBO), or academia such as SWETO-DBLP [2], as well as general-purpose ontologies like TAP [16]. These agreed upon ontologies can then be published to be available for use by other members of the domain.

Building ontologies can be accomplished in one of two approaches: it can start from scratch [9], or it can be built on top of an existing ontology [14]. In both cases, techniques for evaluating the resulting ontology are necessary [11]. Such techniques would not only be useful during the ontology engineering process [10], they can also be useful to an end-user who needs to

find the most suitable ontology among a set of ontologies.

These techniques will be particularly useful in domains where large ontologies including tens of classes and tens of thousands of instances are common. For example, a researcher in the bioinformatics domain who is looking for an ontology that is mainly concerned with genes might have access to many ontologies (e.g. MGED [31], GO, OBO) that cover very similar areas, making it difficult to simply glance through these ontologies to determine the most suitable ontology. In such situations, a tool that would provide an insight into the ontology and describe its features in a way that will allow such a researcher to make a well-informed decision on which ontology to use will be helpful.

In [29] we introduce OntoQA as a suite metrics that evaluate the content of ontologies through the analysis of their schemas and instances in different aspects such as the distribution of classes on the inheritance tree of the schema, the distribution of class instances, and the connectivity between instances of different classes. In this paper, OntoQA ranks ontologies related to a user supplied set of terms. We also refine the metrics introduced previously and add a number of new metrics that help in a better ontology evaluation.

It is important to highlight that ontology features largely depend on the domain the ontology is modeling, therefore, OntoQA allows users to bias the ranking so ontologies that possess certain characteristics (e.g. ontologies with inheritance-only relationships, or deep ontologies) are ranked higher.

Thus, our contributions in this paper can be summarized as the following:

- A flexible technique to rank ontologies based on their contents and their relevance to a set of keywords as well as user preferences.
- According to our knowledge OntoQA is the first approach that evaluates ontologies using their instances (i.e. populated ontologies) as well as schemas.

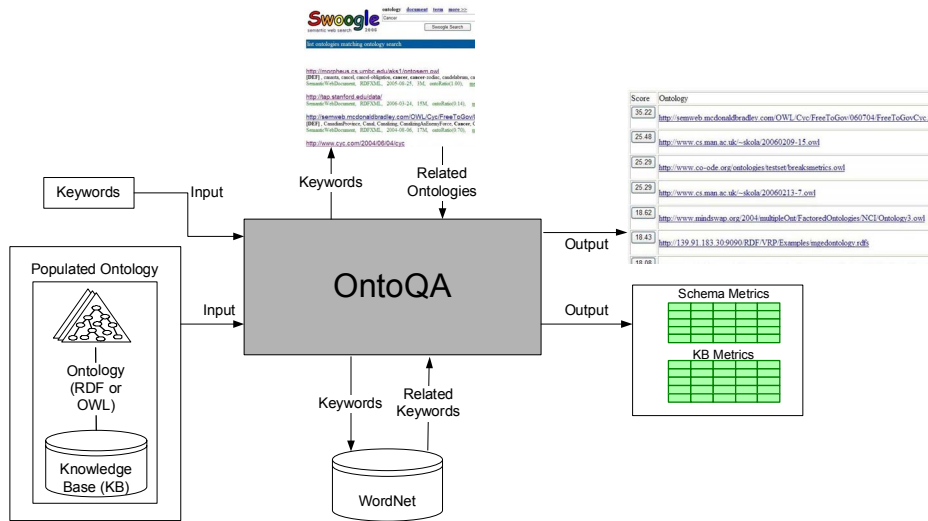


Figure. 1. OntoQA Architecture

2. Architecture

OntoQA was implemented as a public Java web application¹ that uses Sesame [7] as an RDF repository, Figure 1 shows the overall structure of OntoQA. Depending on the input, there are three scenarios for using OntoQA. Here is a step-by-step explanation of how the different OntoQA components are utilized in each case:

1. Ontology:
 - a. OntoQA calculates metric values.
2. Ontology and keywords:
 - a. OntoQA calculates metric values.
 - b. OntoQA uses WordNet [20] to expand the keywords to include any related keywords that might exist in the ontology.
 - c. OntoQA uses the metric values to obtain a numeric value that evaluates the overall contents of the ontology and its relevance to the keywords.
3. Keywords:
 - a. OntoQA uses Swoogle to find the RDF and OWL ontologies in the top 20 results returned by Swoogle.
 - b. OntoQA then evaluates each of the ontologies as indicated in case 2 above.
 - c. OntoQA finally displays the list of ontologies ranked by their score.

3. Terminology

In [29] a terminology was introduced and used in the metric evaluation formulas, here we highlight the main elements of the terminology. The schema of an ontology consists of the following main elements:

- A set of classes, C .
- A set of relationships, P .
- An inheritance function, H^C .
- A set of class attributes, Att .

The knowledgebase of an ontology consists of the following main elements:

- A set of instances, I .
- A class instantiation function, $inst(C_i)$.
- A relationship instantiation function, $instr(I_i, I_j)$.

In addition to the above terms, we introduce the following terms that will be used in the following section:

- The set of class-ancestor pairs in the ontology, $H := \{(C_i, C_j), \text{ where } i \neq j\}$.
- The set of class-ancestor pairs in the inheritance subtree rooted at C_i : $H(C_i) := \{(C_j, C_i), \text{ where } i \neq j \text{ and } H^C(C_j, C_i)\}$
- The set of subclasses of a class C_i : $SubCls(C_i) = \{C_j, \text{ where } H^C(C_j, C_i)\}$.
- The set of relationships a class C_i has with another class C_j : $CREL(C_i) := \{\cup P(C_i, C_j)\}$.
- The set of *distinct* relationships used by instances of a class C_i : $IREL(C_i) := \{\cup instr(I_i, I_j), \text{ where } I_i \in inst(C_i)\}$.

¹ <http://128.192.251.199:8000/OntoQA/>

- The number of *all* relationships used by instances of a class C_i : $SIREL(C_i) := \{\sum |instr(I_i, I_j)|, \text{ where } I_i \in inst(C_i)\}$.
- The set of non-empty classes in the ontology: $C' := \{C_i, \text{ where } inst(C_i) \neq \emptyset\}$.
- The number of instances of a class C_i as expected by the user: $Expected(C_i)$.

4. The Metrics

We divide the evaluation of an ontology on two dimensions: Schema and instances. The first dimension evaluates ontology design and its potential for rich knowledge representation. The second dimension evaluates the placement of instance data within the ontology according to the knowledge modeled in the schema.

In the following sections we will define *metrics* to evaluate each of the above dimensions. These metrics are intended to evaluate certain aspects of ontologies and their potential for knowledge representation.

4.1. Schema Metrics

The schema metrics address the design of the ontology schema. Although it is difficult to know if the ontology design correctly models the knowledge of the domain it is trying to represent, we provide some metrics that indicate different features of an ontology schema.

Relationship Diversity: This metric reflects the diversity of relationships in the ontology. An ontology that contains mostly inheritance relationships (taxonomy) usually conveys less information than an ontology that contains a diverse set of relationships. However, in some applications, users might be interested in ontologies with mostly inheritance relationships (e.g. species classification), and OntoQA gives the user the option to specify whether she prefers a taxonomy or an ontology with diverse relationships.

Definition 1: The relationship diversity (RD) of a schema is defined as the ratio of the number of non-inheritance relationships (P), divided by the total number of relationships defined in the schema (the sum of the number of inheritance relationships (H) and non-inheritance relationships (P)).

$$RD = \frac{|P|}{|H| + |P|}$$

For example, if an ontology has an RD value close to 0 that would indicate that most of the relationships are inheritance relationships. In contrast, an ontology with a value close to 1 would indicate that most of the relationships are non-inheritance.

Schema Deepness: This measure describes the distribution of classes across different levels of the ontology inheritance tree. This measure can distinguish a shallow ontology from a deep ontology. A shallow ontology is an ontology that has a small number of inheritance levels, and each class has a relatively large number of subclasses. In contrast, a deep ontology contains a large number of inheritance levels where classes have a small number of subclasses

Definition 2: The schema depth of the schema (SD) is defined as the average number of subclasses per class.

$$SD = \frac{|H|}{|C|}$$

An ontology with a low SD would be deep, which indicates that the ontology covers a specific domain in a detailed manner (e.g. ProPreO [27]), while an ontology with a high SD would be a shallow (or horizontal) ontology (e.g. TAP), which indicates that the ontology represents a wide range of general knowledge with a low level of detail.

4.2. Instance Metrics

The way instances are placed within an ontology is also a very important aspect of ontology evaluation. The placement of instance data and distribution of the data can indicate the effectiveness of the ontology design and the amount of knowledge represented by the ontology. Instance metrics can be divided on three main sub-dimensions: Overall KB (knowledgebase) metrics that evaluates the overall placement of instances with regard to the schema, class-specific metrics that evaluate the instances of a specific class and compare it to instances of other classes, and relationship-specific metrics that evaluate the instances of a specific relationship and compare it to instances of other relationships.

4.2.1 Overall KB Metrics

This group of metrics gives an overall view on how instances are represented in the KB.

Class Utilization: This metric reflects how classes defined in the schema are being utilized in the KB. This metric can be used to differentiate between two ontologies having the same classes defined in their schemas but one of them populates more classes than the other one, indicating a richer KB.

Definition 3: The class utilization (CU) of an ontology is defined as the ratio of the number of populated classes (C') divided by the total number of classes defined in the ontology schema (C).

$$CU = \frac{|C|}{|C|}$$

The result will be a percentage indicating how the KB utilizes classes defined in the schema. Thus, if the KB has a very low CU, then the KB does not have data that exemplifies all the knowledge that exists in the schema. This metric will be very useful in situations where instances are being extracted into an ontology and it is needed to evaluate the results of the extraction process.

Cohesion: This metric represents the number of connected components in the KB. This metric can particularly help if “islands” form in the KB as a result of extracting data from separate sources that do not have common knowledge, giving insight into what areas need more instances in order to enable the different connected components to connect to each other. Having less connected components (ideally 1) can be helpful, for example, in finding more useful semantic-associations [3] in the ontology.

Definition 4: The cohesion (*Coh*) of an ontology is defined as the number of connected components (*CC*) of the graph representing the KB.

$$Coh = |CC|$$

The result will be an integer representing the number of connected components in the ontology.

Class Instance Distribution: This metric is also useful to evaluate the instance extraction process. It provides an indication on how instances are spread across the classes of the schema. It can be used to discover problems in the instance extraction process.

Definition 5: The class instance distribution of an ontology is defined as the standard deviation in the number of instances per class.

$$CID = StdDev(Inst(C_i))$$

4.2.2 Class-Specific Metrics

This group of metrics indicates how each class defined in the ontology schema is being utilized in the KB.

Class Connectivity: This metric gives an indication of the centrality of a class. With the importance metric mentioned below, both metrics provide a better understanding of how focal some classes are in the KB, which might be help in cases where a user has two ontologies with the similar classes defined in their schemas, but classes that are be important to the user play a central role in one of them, while being on the boundary in the other.

Definition 6: The connectivity of a class (*Conn(C_i)*) is defined as the total number of relationships instances of the class have with instances of other classes.

$$Conn(C_i) = |NIREL(C_i)|$$

Class Importance: This metric is important because it helps in identifying which areas of the schema are in focus when the instances are extracted and inform the user of the suitability of his/her intended use. It will also help direct the ontology developer or data extractor to where s/he should focus on getting data if the intention is to get a consistent coverage of all classes in the schema. Although this measure doesn't consider the real world semantics, where some classes naturally have more instances than others, the class importance can still be used (together with the class connectivity measure mentioned above) to give an indication on what parts of the ontology are considered focal and what parts are on the edges.

Definition 7: The importance of a class (*Imp(C_i)*) is defined as the number of instances that belong to the inheritance subtree rooted at *C_i* in the KB (*inst(C_i)*) compared to the total number of class instances in the KB (*CI*).

$$Imp(C_i) = \frac{|Inst(C_i)|}{|KB(CI)|}$$

Relationship Utilization: This metric reflects how the relationships defined for each class in the schema are being used at the instances level. It is a good indication of the how well the extraction process performed in the utilization of information defined at the schema level. This metric can be used to distinguish between two ontologies having similar schemas but one of them utilizes only a few of the available relationships while other utilizes more.

Definition 8: The relationship richness (*RU*) of a class *C_i* is defined as the number of relationships that are being used by instances *I_i* that belong to *C_i* (*P(I_i,I_j)*) compared to the number of relationships that are defined for *C_i* at the schema level (*P(C_i,C_j)*).

$$RU(C_i) = \frac{|IREL(C_i)|}{|CREL(C_i)|}$$

4.2.3 Relationship-Specific Metrics

This group of metrics indicates how each relationship defined in the ontology schema is being utilized in the KB.

Relationship Importance: This metric measures the percentage of instances of a relationship with respect to the total number of relationship instances in the KB. This metric is important in that it will help in identifying which schema relationships were in focus when the instances were extracted and inform the user of the suitability of his/her intended use. This metric can also help in directing the instance extraction

process to include a more diverse set of relationships the KB doesn't include the required diversity.

Definition 9: The importance of a relationship ($Imp(R_i)$) is defined as the number of instances of relationship R_i in the KB ($inst(R_i)$) compared to the total number of property instances in the KB (R).

$$Imp(R_i) = \frac{|Inst(R_i)|}{|KB(R)|}$$

The result of the formula will be a percentage representing the importance of the current class.

5. Ontology Score Calculation

If the user is searching for ontologies related to a set of terms or is trying to evaluate an ontology regarding a set of terms, OntoQA evaluates the ontology based on the entered keywords in the following manner:

1. The terms entered by the user are extended by adding any related terms obtained using WordNet.
2. OntoQA determines the classes and relationships whose names contain any term of the extended set of terms.
3. OntoQA finally aggregates the schema, the overall KB metrics, and the metrics for all the related classes and relationships to get an overall score for the ontology.

Definition 15: The score of an ontology can be measured as the weighted average of schema metrics, overall KB metrics, and the metrics of related classes and relationships.

$$Score = \sum W_i * Metric_i$$

Where:

$Metric\{\} = \{RD, SD, CU, Coh, \#Classes, \#Relationships, \#Instances, Avg(Conn(C_i)), Avg(Imp(C_i)), Avg(RU(C_i)), Avg(Imp(R_i))\}$ is the set of metrics used in calculating the overall score of an ontology (the averages are for classes and relationships related to the keywords).

$W\{\}$ is the set of weights for each metric.

Please note that the initial values for the set of weights W was set based on empirical testing and can be adjusted with more testing. Also, these weights can be modified by the user to reflect his favor of certain aspects of the ontology.

Among the metrics used to compute the overall score, the relationship diversity (inheritance vs. diverse relationships) and the class deepness (shallow vs. deep ontologies) can be biased towards either option based on the user preference. The other metrics such as the class utilization, connectivity, and importance metrics are always preferred to be higher in better ontologies.

The overall score reflects the overall nature of the ontology and how much it relates to the keywords.

6. Experiments and Evaluation

To illustrate the effectiveness of OntoQA in ranking ontologies, we compare the ranking of the same ontologies by OntoQA, Swoogle, and a group of expert users. We also compare our results with AKTiveRank [1] (presented in Section 7), which is one of the most comparable ranking approaches, using Pearson's Correlation Coefficient.

Table 1 shows the top nine RDF and OWL ontologies ranked by Swoogle when searched for the term "Paper". Each ontology is given a Roman numeral that will be used as a reference to the ontology in other figures. Note that inaccessible ontologies returned by Swoogle are eliminated from this list.

Table 1. Results ranked by Swoogle

Symbol	Ontology URL
I	http://ebiquity.umbc.edu/ontology/conference.owl
II	http://kmi.open.ac.uk/semanticweb/ontologies/owl/aktive-portal-ontology-latest.owl
III	http://www.architexturez.in/+/-c-/caad.3.0.rdf.owl
IV	http://www.csd.abdn.ac.uk/~cmckenzi/playpen/rdf/akt_ontology_LITE.owl
V	http://www.mindswap.org/2002/ont/paperResults.rdf
VI	http://owl.mindswap.org/2003/ont/owlweb.rdf
VII	http://139.91.183.30:9090/RDF/VRP/Examples/SWPG.rdfs
VIII	http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl
IX	http://www.mindswap.org/2004/SSSW04/aktive-portal-ontology-latest.owl

The same term is used in OntoQA producing the results shown in Figure 2. In this figure, the contribution of each metric in the overall score is depicted with different regions in the column for each ontology, and weights are assigned to give a more balanced contribution to each metric, whereas Figure 3 presents results that are biased towards favoring larger ontology schemas.

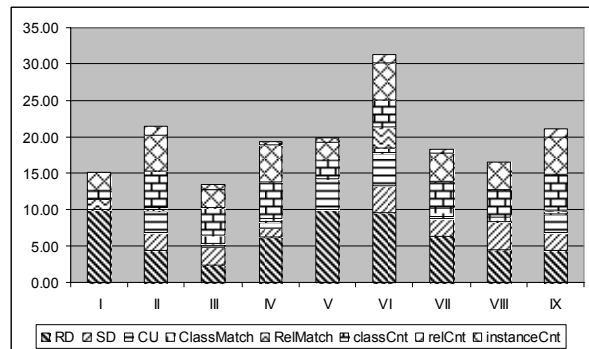


Figure 2. OntoQA results with balanced weights

In Figure 2, ontology VI is ranked the highest. This ontology has a set of 62 rich relationships between its 22 classes, an average of 3 subclasses per parent class, and almost half of the classes are populated. It also has

12 relationships that are related to papers (e.g. author, published in, abstract, etc). All these facts contribute to give this ontology the highest rank.

The differences between OntoQA’s and Swoogle’s rankings are obvious in the figure. The main reason for this difference is that Swoogle follows the OntoRank approach that is similar to Google’s PageRank approach [22], which gives preference to “popular” ontologies. On the other hand, OntoQA ranks ontologies according to their quality measured by the different metrics tuned by users according to their preferences.

A problem with Swoogle’s approach is that if the two copies of the same ontology are placed in two different locations and one of these locations is cited more than the other, Swoogle will rank the copy at this popular location higher than the other copy, even though their contents are the same, while OntoQA will give both ontologies the same ranking.

To further evaluate our approach, the same set of ontologies was ranked by two graduate students in our research lab who are not related to OntoQA and have a longtime experience in building and populating very large scale ontologies (e.g. SWETO-DBLP). These users ranked the ontologies with no relationship to a particular application, which resulted in considering ontologies with larger schemas (number of classes and relationships) as better than ontologies with smaller schemas, even if they were richer ontologies. Their ranking results are shown in Table 2.

Table 2. Results ranked by users

Ontology	Rank	Ontology	Rank	Ontology	Rank
I	9	IV	6	VII	2
II	1	V	8	VIII	7
III	5	VI	4	IX	3

To capture their preferences we re-run our experiment after setting metric weights (W_i) so ontologies with larger schemas are ranked higher, producing the results in Figure 3. Note that other users with particular applications in mind may have different preferences than the expert users in our experiment. Therefore, OntoQA provides flexibility in allowing users with different needs to find ontologies that match their specific needs.

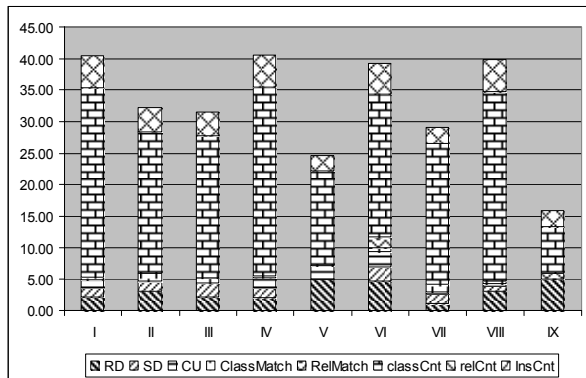


Figure 3. OntoQA results with higher weight for schema size

In this experiment, ontology IV is ranked highest due to its larger schema size (60 classes and 81 relationships). We compare the results in Figure 3 with the user ranking in Table 2. Pearson’s Correlation Coefficient between the two ranking results is 0.80 indicating a relatively high correlation. When AKTiveRank is used in a similar situation in [1], Pearson’s Correlation Coefficient is 0.54 according to our calculation, indicating that the results of OntoQA reflect users’ rankings better. We attempted to run AKTiveRank using the same term that was used here but at the time of writing this paper, it was not available publicly. We were also unable to run the same test that was presented in [1] because out of 12 ontologies used in the test (see Table 1 in [1]); only three were available at the time of writing this paper.

7. Related Work

The increasing interest in the semantic web in recent years resulted in creating a large number of ontologies, and an increasing amount of research is under work on techniques for ontology evaluation. An emerging trend in ontology evaluation is tracking the evolution of ontologies through time. For example, the approach in [24] keeps track of ontology concept evolution through keeping a track of the changes in a version log that can be used to create “virtual versions”. The approach also defines a new language Change Definition Language (CDL) that is used in keeping track of the version. The logical approach in [17] goes even further to discover and repair inconsistencies in ontologies across the different versions of the ontology.

A rule-based approach to conflict detection in ontologies is introduced in [4]. In this approach users define what they consider as conflicting rules using RuleML [6] and the approach will then list any cases

were these rules are violated. A similar approach has also been used in [13].

In [19], the authors propose a complex framework consisting of 160 characteristics spread across five dimensions: content of the ontology, language, development methodology, building tools, and usage costs. Unfortunately, the use of the OntoMetric tool introduced in the paper is not clearly defined, and the large number of characteristics makes their model difficult to understand.

[23] uses a logic model to detect unsatisfiable concepts and inconsistencies in OWL ontologies. The approach is intended to be used by ontology designers to evaluate their work and to indicate any possible problems.

In [28] the authors propose a model for evaluating ontology schemas. The model contains two sets of features: quantifiable and non-quantifiable. It crawls the web (causing some delay, especially if the user has some ontologies to evaluate), searches for suitable ontologies, and then returns the ontology schemas' features to allow the user to select the most suitable ontology for the application. The application does not consider ontologies' KBs' that can provide more insight into the way the ontology is used.

The OntoClean approach in [15] is used for the analysis of taxonomic relationships based on the philosophical notions of rigidity, unity, dependence, and identity.

AKTiveRank authors in propose a set of four metrics to rank a group of ontologies related to a set of terms. The metrics are: class match, density, semantic similarity, and betweenness. These four metrics deal with classes that match the search terms in the ontology. The approach then uses a weighted average of the four metrics to produce a rank for each ontology.

Finally, [8] introduces the ODEval tool that can be used that can be used to detect possible taxonomical problems in ontologies, such as inconsistency, incompleteness, and redundancy.

Table 3. Comparison between different approaches

Approach	User Involvement	Ontologies	Schema / KB
[24]	High	Entered	Schema
[17]	High	Entered	Schema
[4]	High	Entered	Schema + KB
[23]	Low	Entered	Schema
[19]	High	Entered	Schema
[28]	Low	Crawled	Schema
[1]	Low	Crawled	Schema
[8]	Low	Entered	Schema
[15]	Low	Entered	Schema
OntoQA	Low	Crawl + Enter	Schema + KB

Table 3 summarizes some of the main features of the all these approaches. In the user involvement column, it can be seen that the approaches are divided in half in the level of the user involvement required for the approach to successfully achieve its goals. For example, a person using the approach of [24] needs to create a log for each change of the ontology to evaluate any potential problems in the ontology introduced by the change. The second column indicates whether the approach's input ontologies are manually entered by the user or searched for by crawling the internet. The last column indicates whether the approach evaluates the ontology schema only or both the schema and knowledgebase of the ontology.

8. Conclusions and Future Work

In this paper, we present an enhanced version of OntoQA that ranks populated ontologies using a rich set of metrics and by their relation to a set of keywords. OntoQA is different from other approaches in that it is tunable, requires minimal user involvement, and considers both the schema and the instances of a populated ontology.

We plan on using BRAHMS [18] instead of Sesame as a data store since BRAHMS is more efficient in handling large ontologies that are common in bioinformatics. We also plan to enable the user to specify an ontology library (e.g. OBO) to limit the search in ontologies that exist in that specific library.

Acknowledgments

This work is funded by NSF-ITR-IDM Award#0325464 titled 'SemDIS: Discovering Complex Relationships in the Semantic Web' and NSF-ITR-IDM Award#0219649 titled 'Semantic Association Identification Knowledge Discovery for National Security Applications.

References

- [1] Alani H., Brewster C. and Shadbolt N. Ranking Ontologies with AKTiveRank. 5th International Semantic Web Conference. November, 5-9, 2006.
- [2] Aleman-Meza, B., Hakimpour, F., Arpinar, I.B., Sheth A.P. SwetoDblp Ontology of Computer Science Publications, Journal of Web Semantics (2007), doi:10.1016/j.websem.2007.03.001.
- [3] Anyanwu K. and Sheth A. ρ -Queries: Enabling Querying for Semantic Associations on the Semantic Web, Proceedings of the 12th Intl. WWW Conference, Hungary, 2003.
- [4] Arpinar, I.B., Giriloganathan, K., and Aleman-Meza, B Ontology Quality by Detection of

- Conflicts in Metadata. In Proceedings of the 4th International EON Workshop. May 22nd, 2006.
- [5] Berners-Lee T., Hendler J. and Lassila O. The Semantic Web, A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American. 2001.
- [6] Boley H., Tabet S. and Wagner G. Design Rationale of RuleML: A Markup Language for Semantic Web Rules. In the first Semantic Web Working Symposium. Stanford University, California, USA, 2001.
- [7] Broekstra J., Kampman A. and van Harmelen F. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. Proceedings of 1st ISWC, June 9-12th, 2002, Sardinia, Italy.
- [8] Corcho O., Gómez-Pérez A., González-Cabero R., and Suárez-Figueroa M.C. ODEval: a Tool for Evaluating RDF(S), DAML+OIL, and OWL Concept Taxonomies. Proceedings of the 1st IFIP AIAI Conference. Toulouse, France.
- [9] Cristani, M., Cuel, R. A Survey on Ontology Creation Methodologies. International Journal of Semantic Web and Information Systems (IJSWIS), Vol. 1, Issue 2.
- [10] Paslaru P. et al. ONTOCOM: A Cost Estimation Model for Ontology Engineering. Proceedings of fifth ISWC, Athens, GA, USA. November, 2006.
- [11] Fernández M., Gómez-Pérez A., Pazos J., Pazos A. Building a chemical ontology using MethOntology and the ontology design environment. IEEE Intelligent Systems Applications 1999; 4(1):37-45
- [12] Finin T., et al. Swoogle: Searching for knowledge on the Semantic Web. In proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI 05) Pittsburg, Pennsylvania
- [13] Friedrich G. and Shchekotykhin K. A General Diagnosis Method for Ontologies. In Proceedings of the 4th International Semantic Web Conference (ISWC05), pages 232-246, 2005.
- [14] Gómez-Pérez A., Rojas-Amaya M. Ontological Reengineering for Reuse. Proceedings of the 11th European Workshop on Knowledge Acquisition, Modeling and Management.
- [15] Guarino N. and Welty C. Evaluating Ontological Decisions with OntoClean. Communications of the ACM, 45(2) 2002, pp. 61-65
- [16] Guha R. and McCool R.: TAP: A Semantic Web Test-bed. Journal of Web Semantics, 2003.
- [17] Haase P., van Harmelen F., Huang Z., Stuckenschmidt H., and Sure Y. A framework for handling inconsistency in changing ontologies. In Proceedings of ISWC2005, 2005.
- [18] Janik M., Kochut K. BRAHMS: A WorkBench RDF Store and High Performance Memory System for Semantic Association Discovery. Fourth International Semantic Web Conference, Galway, Ireland, 6-10 November 2005
- [19] Lozano-Tello A. and Gomez-Perez A. ONTOMETRIC: a method to choose the appropriate ontology. Journal of Database Management 2004.
- [20] Miller. G. WordNet: A lexical database for english. Communications of the ACM, vol. 38, no. 11, 1995.
- [21] OWL: Web Ontology Language Overview, W3C Recommendation, February 2004. (<http://www.w3.org/TR/owl-features/>).
- [22] Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1998. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Database group.
- [23] Parsia B., Sirin E. and Kalyanpur A. Debugging OWL Ontologies. Proceedings of WWW 2005, May 10-14, 2005, Chiba, Japan.
- [24] Plessers P. and De Troyer O. Ontology Change Detection Using a Version Log. In Proceedings of the 4th ISWC, 2005.
- [25] RDF Schema, W3C Recommendation, February 2004. (<http://www.w3.org/TR/rdf-schema/>)
- [26] RDF: Resource Description Framework, W3C Recommendation, February 2004. (<http://www.w3.org/TR/2004/REC-rdf-primer-20040210>).
- [27] Sahoo S. et al. Knowledge Modeling and its Application in Life Sciences: A Tale of two Ontologies. The 15th WWW Conference. Edinburgh, Scotland, United Kingdom. 2006.
- [28] Supekar K., Patel C. and Lee Y. Characterizing Quality of Knowledge on Semantic Web. Proceedings of AAAI FLAIRS, May 17-19, 2004, Miami Beach, Florida.
- [29] Tartir S. et al. OntoQA: Metric-Based Ontology Quality Analysis. Proceedings of IEEE ICDM 2005 KADASH Workshop.
- [30] The Gene Ontology. <http://www.geneontology.org>
- [31] The MGED Ontology <http://mged.sourceforge.net/ontologies>
- [32] Open Biomedical Ontologies <http://obo.sourceforge.net/>