OntoQA: Metric-Based Ontology Quality Analysis

Samir Tartir
Mawdoo3.com
15 PUBLICATIONS **481** CITATIONS
SEE PROFILE

Ismailcem Budak Arpinar
University of Georgia
87 PUBLICATIONS **2,702** CITATIONS
SEE PROFILE

Amit Sheth
University of South Carolina
1,008 PUBLICATIONS **32,776** CITATIONS
SEE PROFILE

Boanerges Aleman-Meza
Rice University
77 PUBLICATIONS **2,001** CITATIONS
SEE PROFILE

OntoQA: Metric-Based Ontology Quality Analysis

Samir Tartir, I. Budak Arpinar, Michael Moore, Amit P. Sheth, Boanerges Aleman-Meza
LSDIS Lab
Department of Computer Science
University of Georgia
Athens, GA 30602
USA
{tartir, budak, moore, amit, boanerg}@cs.uga.edu

Abstract

As the Semantic Web gains importance for sharing knowledge on the Internet this has lead to the development and publishing of many ontologies in different domains. When trying to reuse existing ontologies into their applications, users are faced with the problem of determining if an ontology is suitable for their needs. In this paper, we introduce OntoQA, an approach that analyzes ontology schemas and their populations (i.e. knowledgebases) and describes them through a well defined set of metrics. These metrics can highlight key characteristics of an ontology schema as well as its population and enable users to make an informed decision quickly. We present an evaluation of several ontologies using these metrics to demonstrate their applicability.

1. Introduction

The Semantic Web envisions making web content machine processable, not just readable or consumable by human beings [3]. This is accomplished by the use of ontologies which involve agreed terms and their relationships in different domains (e.g., the gene ontology (GO) and other ontologies at Open Biology Ontologies[1] in biology as well as general-purpose ontologies such as SWETO – Semantic Web Technology Evaluation Ontology [1], and TAP[6]). Different users can agree on the use of a common ontology in RDF(S) (Resource Description Framework) [15, 16] or OWL (Web Ontology Language) [12] in order to annotate their content or resolve their differences through interactions and negotiations (i.e. emergent semantics).

An ontology describes a hierarchy of concepts usually related by subsumption relationships. In more sophisticated cases, suitable axioms are added in order to express other relationships between concepts and to constrain their intended interpretation [5]. After an ontology is constructed, it is usually populated by instances either manually, semi-automatically or mostly-automatically. For example, the IMPs architecture [22] and SCORE [21] facilitate the retrieval, crawling, extraction disambiguation, restructuring, integration and formalization of task-relevant ontological knowledge from the semi-structured and structured sources on web.

Assessing the quality of an ontology is important for several reasons including allowing the ontology developer to automatically recognize areas that might need more work, allowing the ontology user to know what parts of the ontology might cause problems, and allow him/her to compare between different ontologies when only one is going to be used.

In our view, the quality of ontologies can be assessed in different dimensions. For example, quality metrics can be used to evaluate the success of a schema in modeling a real-world domain such as computer science researchers and their publications (Quality 1 in Figure 1). The depth, breadth, and height balance of the schema inheritance tree can play a role in a quality assessment. Additionally, the quality of a populated ontology (i.e., KB) can be measured to check whether it is a rich and accurate representative of real world entities and relations (Quality 2 in Figure 1). Finally, the quality of KB can be measured to see if the instances and relations agree with the schema (Quality 3 in Figure 1).

We propose a method to evaluate the quality of an ontology on the different dimensions mentioned above. This method can be used by ontology users before considering an ontology as a source of information or by ontology developers to evaluate their work in building the ontology.

[1] http://obo.sourceforge.net

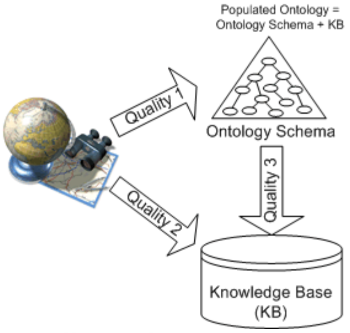


Fig. 1. Different dimensions to evaluate ontology quality

Our contributions in this paper are the following:

- Categorizing the quality of ontologies into three groups: schema, knowledgebase (KB) and class metrics. These metrics serve as a means to evaluate the quality of a single ontology or to compare ontologies when more than one candidate fits certain requirements.
- Providing metrics to quantitatively assess the quality in each group.
- A tool for quality analysis and providing experimental results.

The rest of the paper is organized as follows: Section 2 provides the motivation to our work. Section 3 details the model we base our work on. Section 4 presents the metrics of our model. Section 5 discusses the implementation and presents experimental results. Finally, Section 6 discusses previous related work and compares that work to our approach.

2. Motivation

The motivation for our work began during our work on the SWETO ontology [1]. SWETO is intended to be a broad and general purpose ontology covering multiple domains and populated with real data from heterogeneous sources. One purpose of not limiting SWETO to a single domain enabled us to harvest facts from open-source, non-copyrighted Web sources to populate it with approximate one million facts. We wanted it to serve as a test-bed for advanced semantic applications such as discovery of semantic associations and semantic entity disambiguation in our own research, as well as to make it available to the research community for scalability and performance testing of techniques at the RDF/S level. Semantic associations [2] are the paths (entities and relationships) that connect two different entities. The nature of SWETO requires the careful design of the schema and the extraction of data from a large number of distinct resources to cover the different schema classes in such a way that represents the real world.

SWETO includes some geographical data represented by classes of cities, states, and countries. It also contains information about logistic and financial aspects of terrorism. The publications domain is included in SWETO by adding classes representing Researchers, Scientific Publications, Journals, Conferences and Books. SWETO also includes information about business organizations such as Companies and Banks.

The extraction process was done mostly-automatically on several phases and resulted in hundreds of thousands of instances in the KB. Some of the sources used were the CIA World Factbook[2], which includes rich geographical information, and conference web sites. After each phase of the extraction process, there was a need to evaluate the quality of the extracted data and to decide on the targets for the next extraction phase. Some of the issues that needed additional attention included the abundance of instances on some parts of the schema while other parts have no instances, and that instances of some classes are focusing on using some of the relationships defined in the schema while ignoring the other relationships, as will be shown in the experimental results in Section 5. These problems can result in the lack of rich semantic associations in the SWETO KB; restricting relationships found between two persons to co-authorship in a certain publication, where a more interesting relationship that was not captured due to not being extracted that can establish business interests between these two persons.

The discovery of these and similar problems is a difficult process because of the large number of classes in the schema and the large number of instances that belong to these classes. The set of metrics presented here can be used to describe an ontology's schema and KB to provide the ontology designer with information they can use to further enhance such ontology. These metrics can be used not only during the development of ontologies but also by a user looking for an ontology to suit his/her needs to compare between different existing ontologies.

3. Model

OntoQA is used to describe different metrics of an ontology using the vocabulary defined in an RDF-S or OWL document and instances defined in an RDF file, requiring no further information in all metrics (with the exception of the metric that requires information about the expected number of instances for each class). The model considers how classes are organized in the schema and on how instances are distributed across the schema.

The model that will be used in the definition of the metrics is based on [8]. It formally defines the schema

[2] http://www.cia.gov/cia/publications/factbook

and KB structures. This model is going to be used in the definition of metrics in Section 5.

Ontology structure (Schema). An ontology schema is a 6-tuple $O := \{C, P, A, H^C, prop, att\}$, consisting of two disjoint sets C and P whose elements are called concepts and relationships, respectively, a **concept hierarchy** H^C : H^C is a directed, transitive relation $H^C \subseteq C \times C$ which is also called concept taxonomy. $H^C(C_1, C_2)$ means that C_1 is a sub-concept of C_2, a **function** $prop$: $P \rightarrow C \times C$, that relates concepts non-taxonomically (The function dom: $P \rightarrow C$ *with* $dom(P) := \prod_1(rel(P))$ gives the domain of P, and $range$: $P \rightarrow C$ with $range(P) := \prod_2(rel(P))$ gives its range. For $prop(P) = (C_1, C_2)$ one may also write $P(C_1, C_2)$). A specific kind of relations are attributes A. The **function** att: $A \rightarrow C$ relates concepts with literal values (this means $range(A) := STRING$).

Knowledgebase (metadata) structure. A metadata structure is a *6-tuple* $MD := \{O, I, L, inst, instr, instl\}$, that consists of an ontology O, a set I whose elements are also called instance identifiers (correspondingly C, P and I are disjoint), a set of literal values L, a function $inst: C \rightarrow 2^I$ called **concept instantiation** (For $inst(c) = I$ one may also write $C(I)$), and a function $instr : P \rightarrow 2^{IxI}$ called **relation instantiation** (for $inst(P) = P\{I_1, I_2\}$ one may also write $P(I_1, I_2)$). The **attribute instantiation** is described via the function $instl: P \rightarrow 2^{IxL}$ relates instance s with literal values.

4. The Metrics

The metrics we are proposing are not 'gold standard' measures of ontologies. Instead, the metrics are intended to evaluate certain aspects of ontologies and their potential for knowledge representation. Rather than describing an ontology as merely effective or ineffective, metrics describe a certain aspect of the ontology because, in most cases, the way the ontology is built is largely dependent on the domain in which it is designed. Ontologies modeling human activities (e.g., travel or terrorism) will have distinctly different characteristics from those modeling the natural (or physical) world (e.g. genomes or complex carbohydrates.

We divided the metrics into two related categories: schema metrics and instance metrics. The first category evaluates ontology design and its potential for rich knowledge representation. The second category evaluates the placement of instance data within the ontology and the effective usage of the ontology to represent the knowledge modeled in the ontology.

4.1. Schema Metrics

The schema metrics address the design of the ontology. Although we cannot know if the ontology design correctly models the knowledge, we can provide metrics that indicate the richness, width, depth, and inheritance of an ontology schema.

Relationship Richness: This metric reflects the diversity of relations and placement of relations in the ontology. An ontology that contains many relations other than class-subclass relations is richer than a taxonomy with only class-subclass relationships.

Formally, the relationship richness (*RR*) of a schema is defined as the ratio of the number of relationships (*P*) defined in the schema, divided by the sum of the number of subclasses (*SC*) (which is the same as the number of inheritance relationships) plus the number of relationships.

$$RR = \frac{|P|}{|SC| + |P|}$$

The result of the formula will be a percentage representing how much of the *connections* between classes are rich relationships compared to all of the possible connections that can include rich relationships and inheritance relationships. For example, if an ontology has an RR close to zero, that would indicate that most of the relationships are class-subclass (i.e. ISA) relationships. In contrast, an ontology with a RR close to one would indicate that most of the relationships are other than class-subclass.

Attribute Richness: The number of attributes (slots) that are defined for each class can indicate both the quality of ontology design and the amount of information pertaining to instance data. In general we assume that the more slots that are defined the more knowledge the ontology conveys.

Formally, the attribute richness (*AR*) is defined as the average number of attributes (slots) per class. It is computed as the number attributes for all classes (*att*) divided by the number of classes (C).

$$AR = \frac{|att|}{|C|}$$

The result will be a real number representing the average number of attributes per class, which gives insight into how much knowledge about classes is in the schema. An ontology with a high value for the AR indicates that each class has a high number of attributes on the average, while a lower value might indicate that less information is provided about each class.

Inheritance Richness: This measure describes the distribution of information across different levels of the ontology's inheritance tree or the fan-out of parent classes. This is a good indication of how well knowledge is grouped into different categories and subcategories in the ontology. This measure can distinguish a horizontal ontology from a vertical ontology or an ontology with different levels of specialization. A horizontal (or flat) 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 vertical ontology contains a large number of inheritance levels where classes have a small number of subclasses. This metric can be measured for the whole schema or for a subtree of the schema.

Formally, the inheritance richness of the schema (IR_s) is defined as the average number of subclasses per class. The number of subclasses (C_1)for a class C_i is defined as $\left|H^C(C_1, C_i)\right|$

$$IR_S = \frac{\sum_{C_i \in C} \left|H^C(C_1, C_i)\right|}{|C|}$$

The result of the formula will be a real number representing the average number of subclasses per class. An ontology with a low IR_S would be of a vertical nature, which might reflect a very detailed type of knowledge that the ontology represents. while an ontology with a high IR_S would be of a horizontal nature, which means that ontology represents a wide range of general knowledge.

4.2. Instance Metrics

The way data is placed within an ontology is also a very important measure of ontology quality. 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 are grouped into two categories: KB metrics, which describe the KB as a whole, and Class metrics, which describe the way each class that is defined in the schema is being utilized in the KB.

4.2.1. Knowledgebase Metrics.

Class Richness: This metric is related to how instances are distributed across classes. The number of classes that have instances is compared with the total number of classes, giving a general idea of how many instances are related to classes defined in the schema.

Formally, the class richness (CR) of a KB is defined as the ratio of the number of classes used in the base (C`) divided by the number of classes defined in the ontology schema (C).

$$CR = \frac{|C`|}{|C|}$$

The result will be a percentage indicating how rich in classes the KB is. Thus, if the KB has a very low CR, then the KB does not have data that exemplifies all the knowledge in the schema. On the other hand, a KB that has a very high CR (close to 100%) would indicate that the data in the KB represents most of the knowledge in the schema.

Average Population (average distribution of instances across all classes): This measure is an indication of the number of instances compared to the number of classes. It can be useful if the ontology developer is not sure if enough instances were extracted compared to the number of classes.

Formally, the average population (P) of classes in a KB is defined as the number of instances of the KB (I) divided by the number of classes defined in the ontology schema (C).

$$P = \frac{|I|}{|C|}$$

The result will be a real number that shows how well is the data extraction process that was performed to populate the KB. For example, if the average number of instances per class is low, when read in conjunction with the previous metric, this number would indicate that the instances extracted into the KB might be insufficient to represent all of the knowledge in the schema. Keep in mind that some of the schema classes might have a very low number or a very high number by the nature of what it is representing.

Cohesion: If instances and the relationships among them are considered as a graph where nodes represent instances and edges represent the relationships between them, this metric is the number of separate connected components in the instances. It can be used to indicate what areas need more instances in order to enable instances to be more closely connected. This metric can help if “islands” form in the KB as a result of extracting data from separate sources that do not have common knowledge.

Formally, the cohesion (Coh) of a KB is defined as the number of separate connected components (SCC) of the graph representing the KB.

$$Coh = |SCC|$$

The result will be an integer representing the number of separate components. For example, a more useful throughput of semantic-association discovery algorithms might be expected from an ontology with a Coh of 1 (as this would indicate that all data in the KB is connected, and it will be possible to use a semantic association discovery algorithm without worrying about not considering a part of the KB).

4.2.2. Class Metrics.

Importance: The percentage of instances that belong to classes at the subtree rooted at the current class with respect to the total number of instances. This metric can also be called instance distribution as it refers to the distribution of instances over classes. This metric is important in that it will help 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 towards 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 might not be exact, it can be used to give a clear idea on what parts of the ontology are considered focal and what parts are on the edges.

Formally, the importance (*Imp*) of a class C_i is defined as the number of instances that belong to the subtree rooted at C_i in the KB ($C_i(I)$) compared to the total number of instances in the KB (I).

$$Imp = \frac{|C_i(I)|}{|I|}$$

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

Fullness: This metric details the KB average population metric mentioned above. It would be mainly used by an ontology developer interested in knowing how well the data extraction was with respect to the expected number of instances of each class. This is helpful in directing the extraction process to any resources that will add instances belonging to classes that are not full.

Formally, the fullness (F) of a class C_i is defined as the *actual* number of instances that belong to the subtree rooted at C_i ($C_i(I)$) compared to the *expected* number of instances that belong to the subtree rooted at C_i ($C_i`(I)$).

$$F = \frac{|C_i(I)|}{C_i`(I)}$$

The result of the formula will be a percentage representing the actual coverage of instances compared to the expected coverage. In most cases, this measure is an indication of how well the instance extraction process performed. For example, a KB where most classes have a low F would require more data extraction. On the other hand, a KB where most classes are almost full would indicate that it reflects more closely the knowledge encoded in the schema.

Inheritance Richness: This measure details the schema IR_S metric mentioned above and describes the distribution of information in the current class subtree per class. This measure is a good indication of how well knowledge is grouped into different categories and subcategories under this class.

Formally, the inheritance richness (IR_c) of class C_i is defined as the average number of subclasses per class in the subtree. The number of subclasses for a class C_i is defined as $\left|H^C(C_1, C_i)\right|$ and the number of nodes in the subtree is $|C'|$.

$$IR_C = \frac{\sum_{C_i \in C'} \left|H^C(C_1, C_i)\right|}{|C'|}$$

The result of the formula will be a real number representing the average number of classes per schema level. The interpretation of the results of this metric depends highly on the nature of the ontology. Classes in an ontology that represents a very specific domain will have low IR_C values, while classes in an ontology that represents a wide domain will usually have higher IR_C values.

Relationship Richness: This is an important metric reflecting how much of the properties in each class in the schema is actually 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.

Formally, the relationship richness (*RR*) 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 ($P(C_i,C_j)$).

$$RR_c = \frac{|P(I_i, I_j), I_i \in C_i(I)|}{|P(C_i, C_j)|}$$

The result of the formula will be a percentage representing how well the KB utilizes the knowledge defined in the schema regarding the class in focus. For example, if most classes have low RRC values, this would mean that instances are using only a few number of the class relationships in the schema in contrast to another ontology where instances have relationships that span most of the relationships available at the class level in the schema.

Connectivity: This metric is intended to give an indication of the number of relationships instances of each class to other instances. This measure works in tandem with the importance metric mentioned above to create a better understanding of how focal some classes function. For more details, instances within a class can be grouped based on the number of relationships they have with other instances.

Formally, the connectivity (*Cn*) of a class C_i is defined as the number of instances of other classes that are connected to instances of that class (I_j).

$$Cn = \left|I_j, P(I_i, I_j) \wedge I_i \in C_i(I)\right|$$

The result of the formula will be an integer representing the popularity of instances of the class. A class with a high Cn plays a central role in the ontology compared to a class with a lower value. This measure can be used to understand the nature of the ontology by indicating which classes play a central role compared to other classes.

Readability: This metric indicates the existence of human readable descriptions in the ontology, such as comments, labels, or captions. This metric can be a good indication if the ontology is going to be queried and the results listed to users.

Formally, the readability (*Rd*) of a class C_i is defined as the sum of the number attributes that are comments and the number of attributes that are labels the class has.

$$Rd = |A, A = rdfs:comment| + |A, A = rdfs:label|$$

The result of the formula will be an integer representing the availability of human-readable information for the instances of the current class.

5. Implementation and Experiments

We implemented the metrics presented above in a Java-based prototype. The system first calculates the ontology schema metrics, which is defined using an RDFS or OWL file, and then uses the given RDF file to compute the instance metrics. Our implementation uses the Sesame RDF store [4] to load data for the ontology schema and KB. For a data stored, Sesame and Jena were considered. Finally, Sesame was selected because it was able to handle large data sizes compared with the Jena data store [9].

The main obstacle in experimenting with our model was the lack of ontologies that offer their schema and have a KB of a large size (>1 MB) reflecting the intended use of the schema.

Results of running the application on the following ontologies are discussed below:

1. SWETO. SWETO is our general purpose ontology that covers domains including publications, affiliations, geography and terrorism.
2. TAP [6]. TAP is Stanford's general purpose ontology. It is divided into 43 domains. Some of these domains are publications, sports, and geography.
3. GlycO [20]. GlycO is another ontology under development in the LSDIS Lab for the Glycan Expression. Its goal is to develop a suite of databases in addition to computational tools that facilitate efficient acquisition, description, analysis, sharing and dissemination of the data contained therein.
4.

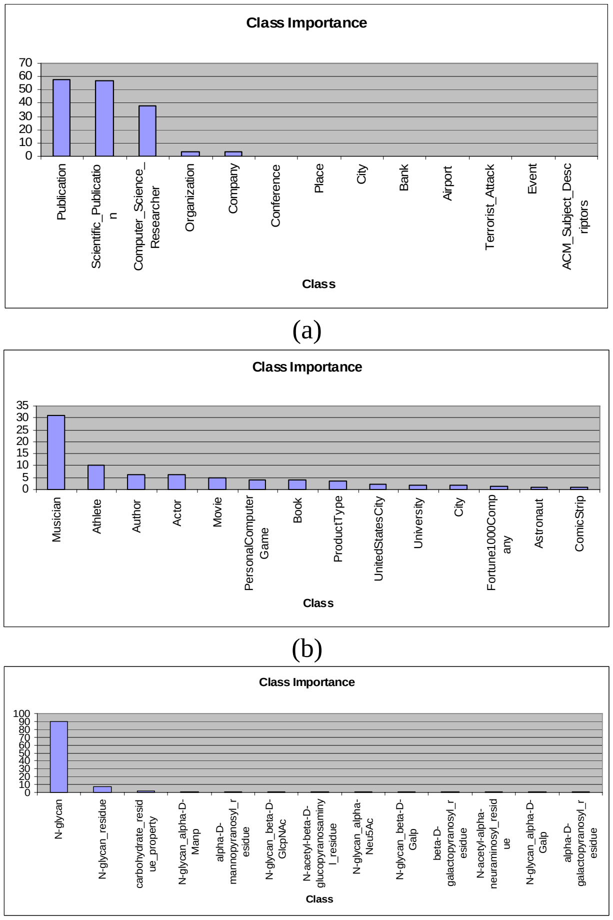
Ontology	Classes	Instances	Inheritance Richness
SWETO	44	813,217	4.00
TAP	3,229	70,850	5.36
GlycO	352	2,034	1.56

Table 1. Summary of SWETO and TAP

The table above shows that TAP is the most general due to the large value for its inheritance richness (fan-out) and the richest ontology in the three with the largest number of instances. GlycO, on the other hand, is clearly domain specific as indicated by its small number of subclasses per class and by its small number of instances with a relatively high number of subclasses per class and a large number of instances, SWETO is somewhere in the middle, and it can be classified as a moderately general purpose ontology.

5.1. Class importance

Using the class importance metric to compare the above three ontologies clearly shows how they are intended to be used. Figure 2 shows the most important classes in each ontology.



(a)

(b)

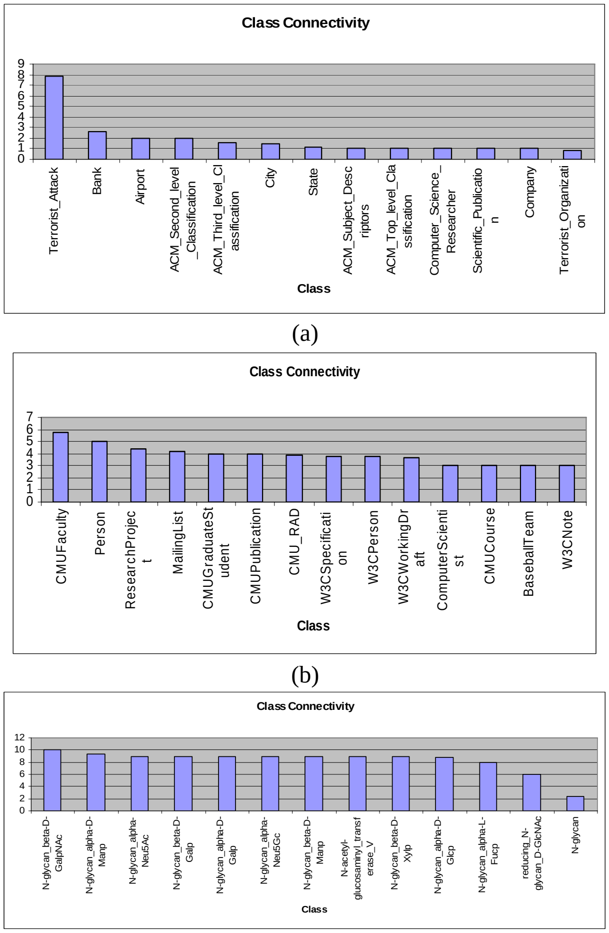
(c)

Fig. 2. Class importance in (a) SWETO (b) TAP and (c) GlycO

From the figure it can be clearly seen that classes related to publications are the dominant classes in SWETO. While, with the exception of the Musician class, TAP gives consistent importance to most of its classes covering the different domains it includes. The nature of the GlycO ontology is reflected in the classes that are most important. The importance of the "N-glycan_residue" and the "alpha-D-mannopyranosyl_residue" and other classes show the narrow domain of GlycO is intended for, although the "glycan_moiety" class is the most important class covering about 90% of the instances in the KB.

5.2. Class connectivity

As explained above, class connectivity is used to indicate which classes play a more central role than other classes, which is another way of describing the nature of an ontology. Figure 3 shows the most connected classes in each of the three ontologies.



(a)

(b)

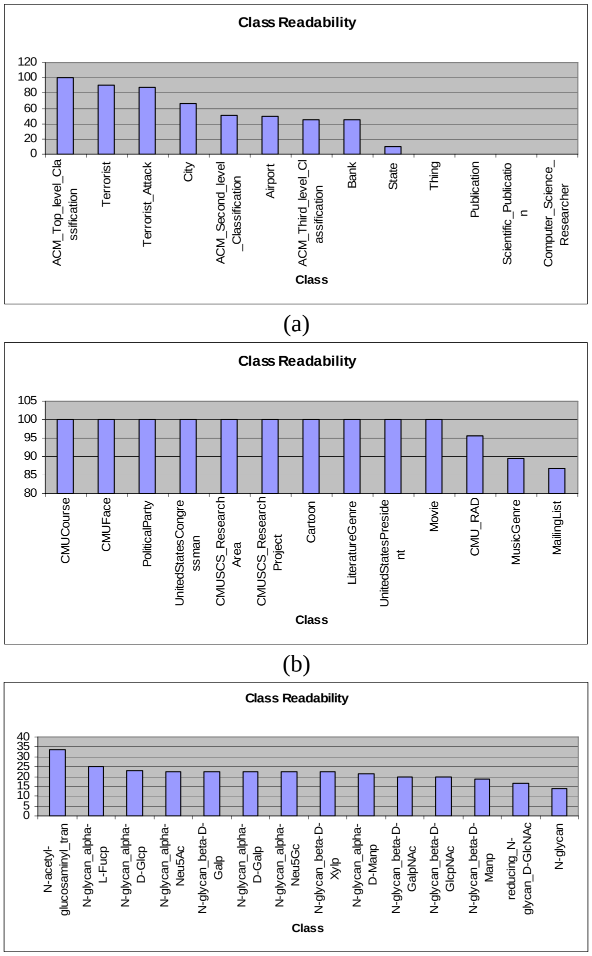
(c)

Fig. 3. Class connectivity in (a) SWETO (b) TAP and (c) GlycO

Figure 3 above shows that SWETO also includes good information about domains other than publications, including the terrorism domain (Terrorist_Attack and Terrorist_Organization), the business domain (Bank and Company) and geographic information (City and State). In a similar manner, TAP continues to show that it covers different domains, and its most connected classes cover the education domain (CMUCourse and CMUSCS_ResearchArea), the entertainment domain (TV and Movie), and other domains as well. GlycO's specific-purpose nature is evident from the Glycan related classes that are most connected.

5.3. Class readability

Class readability is a useful metric when there is an intention to frequently use an ontology by humans. Figure 4 shows the most readable classes in each of the three ontologies.



(a)

(b)

(c)

Fig. 4. Class readability in (a) SWETO (b) TAP and (c) GlycO

With different degrees, all three ontologies include readable information. SWETO does not provide human readable information to most of the classes, which can be a concern if the ontology is going to be used by humans. On the other hand, both TAP and GlycO can be considered human-friendly as they provide descriptive information for most of their classes.

6. Related Work

In recent years, increasing interest has been given to ontology design and quality. In [7], 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.

[10] provides a seven-step guide for developing ontology. The steps include guidelines ranging from what to include in the ontology, how to build a good class hierarchy, how to create class slots (attributes), and finally to populating the KB of the ontology. This guide is intended for developers and would not help users in the evaluation of an existing ontology.

[13] 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 the quality of their work and to indicate any possible problems.

In [19] 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' quality that can provide more insight into the way the ontology is used.

[11] defines a framework for comparing ontology schemas. It compares CYC, Dahlgren's, Generalized upper model, GENSIM, KIF, PLNIUS, Sowa's, TOVE, UMLS, and WORDNET. The framework defines characteristics that can be used to compare these ontologies. These characteristics are divided into the following groups: design process, taxonomy, internal concept structure and relations between concepts, axioms, inference mechanism, applications, and contribution. The authors' goal was a review of current design ontology schema design techniques by manually inspecting them and classifying them into different design categories.

In [18], the authors introduce an environment for ontology development called DODDLE-R. DODDLE-R, which consists of two parts: a pre-processing part that generates a prototype ontology, and a quality improvement part to refine that ontology. The quality improvement part focuses on fixing the problems related to the issue of Concept Drift where positions of particular concepts changes depending on the domain. This approach can be helpful for experts trying to build an ontology from scratch, but it is does not serve users who are not design experts and who only want an ontology that fits their needs.

Table 2 below summarizes the approaches discussed above. It considers the target audience ('D' = 'Developers' and 'E' = 'End Users'), whether the approach is automatic or manual, whether it considers the schema or both the schema and the KB ('S' = 'Schema'), and whether the approach allows the user to specify the ontologies s/he wants to analyze.

Approach	Target	Auto/Man	S/KB	Ontology
OntoMetric	D	Manual	S	Input
OntoDev	D	Manual	S+KB	Input
Swoop	D	Auto	S	Input
Charac	D + E	Auto	S	Crawled
Survey	D	Manual	S	Input
Doddle-R	D	Manual	S	Input
OntoQA	D + E	Auto	S+KB	Input

Table 2. Summary of current ontology quality management approaches

8. Conclusions and Future Work

In this paper, we show how OntoQA can be used to describe ontologies in a way that enables the user or ontology developer determine the quality of an ontology.

We envision future releases of OntoQA to allow the calculation of domain-dependent metrics that make use of some standard ontologies in a certain domain. We are also planning on making OntoQA a web-enabled tool where users can enter their ontology files' path and use our application to measure the quality of the ontology.

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] Aleman-Meza, B., Halaschek, C., Sheth, A., Arpina, I. B., Sannapareddy, G. SWETO: Large-Scale Semantic Web Test-bed. Proceedings of the 16th International Conference on Software Engineering & Knowledge Enginerring (SEKE2004): Workshop on Ontology in Action, Banff, Canada, June 21-24, 2004, pp. 490-493

[2] Anyanwu K. and Sheth, A. -Queries: Enabling Querying for Semantic Associations on the Semantic Web, Proceedings of the 12th Intl. WWW Conference, Budapest, Hungary, 2003.

[3] 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, May 2001.

[4] Broekstra, J., Kampman, A. and van Harmelen, F. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. Proceedings of 1st International Semantic Web Conference (ISWC 2002), June 9-12th, 2002, Sardinia, Italia.

[5] Guarino N. Formal Ontology and Information Systems. Proceedings of FOIS'98, Trento, Italy, 6-8 June 1998. Amsterdam, IOS Press, pp. 3-15.

[6] Guha, R., McCool, R.: TAP: A Semantic Web Test-bed. Journal of Web Semantics, 1(1):81-87, (2003).

[7] Lozano-Tello, A. & Gomez-Perez, A. ONTOMETRIC: a method to choose the appropriate ontology. Journal of Database Management 2004, 15, 1–18.

[8] Maedche, A. and Zacharias, V. Clustering Ontology-based Metadata in the Semantic Web. Proceedings of the 6th European Conference, PKDD 2002, Helsinki, Finland, August 19-23, 2002.

[9] Mc Bride, B. Jena: Implementating the RDF Model and Syntax Specification, Proceedings of the Second International Workshop on the Semantic Web (SemWeb'2001), May 2001.

[10] Noy, N. and McGuinness, D. Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.

[11] Noy, N., and Hafner, C. The state of the art in ontology design: A survey and comparative review. AI Magazine, 18(3):53-74, 1997.

[12] OWL: Web Ontology Language Overview, W3C Recommendation, February 2004. (http://www.w3.org/TR/owl-features/).

[13] Parsia, B., Sirin, E. and Kalyanpur, A. Debugging OWL Ontologies. Proceedings of WWW 2005, May 10-14, 2005, Chiba, Japan.

[14] Patel, C., Supekar, K., Lee, Y. and Park, E.K. OntoKhoj: A Semantic Web Portal for Ontology Searching, Ranking and Classification. Proceedings of 5fth International Workshop on Web Information and Data Management (WIDM'03), November 7–8, 2003, New Orleans, Louisiana, USA.

[15] RDF Schema, W3C Recommendation, February 2004. (http://www.w3.org/TR/rdf-schema/)

[16] RDF: Resource Description Framework, W3C Recommendation, February 2004 (http://www.w3.org/TR/2004/REC-rdf-primer-20040210).

[17] Sheth, A. and Avant, D. Semantic Visualization: Interfaces for exploring and exploiting ontology, knowledgebase, heterogeneous content and complex relationships. NASA Virtual Iron Bird Workshop, March 31 and April 2, California (2004)

[18] Sugiura, N., Shigeta, Y., Fukuta, N., Izumi, N. and Yamaguchi, T. Towards On-the-fly Ontology Construction – Focusing on Ontology Quality Improvement. Proceedings of the First European Semantic Web Symposium, ESWS 2004 Heraklion, Crete, Greece, May 10-12, 2004.

[19] Supekar, K., Patel, C. and Lee, Y. Characterizing Quality of Knowledge on Semantic Web. Proceedings of AAAI Florida AI Research Symposium (FLAIRS-2004), May 17-19, 2004, Miami Beach, Florida.

[20] Sheth, A. et al. Semantic Web technology in support of Bioinformatics for Glycan Expression" W3C Workshop on Semantic Web for Life Sciences, 27-28 October 2004, Cambridge, Massachusetts USA

[21] Sheth, A., Bertram, C., Avant, D., Hammond, B., Kochut, K., Warke, Y. Managing Semantic Content for the Web. IEEE Internet Computing archive Volume 6 , Issue 4 (July 2002)

[22] Crow L., Shadbolt N. Extracting Focused Knowledge From The Semantic Web. International Journal of Human-Computer Studies, Volume 54, Number 1, January 2001, pp. 155-184(30)