The 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference

ISWC+ASWC 2007

Workshop 6

Evaluation of Ontologies and Ontology-based tools

Workshop Organizers:

Denny Vrandecic, Raúl García-Castro,
Asunción Gómez Pérez, York Sure, Zhisheng Huang

11 Nov. 2007
BEXCO, Busan KOREA

ISWC 2007 Sponsor

Emerald Sponsor

Gold Sponsor

Silver Sponsor

We would like to express our special thanks to all sponsors

ISWC 2007 Organizing Committee

General Chairs

Riichiro Mizoguchi (Osaka University, Japan)

Guus Schreiber (Free University Amsterdam, Netherlands)

Local Chair

Sung-Kook Han (Wonkwang University, Korea)

Program Chairs

Karl Aberer (EPFL, Switzerland)

Key-Sun Choi (Korea Advanced Institute of Science and Technology)

Natasha Noy (Stanford University, USA)

Workshop Chairs

Harith Alani (University of Southampton, United Kingdom)

Geert-Jan Houben (Vrije Universiteit Brussel, Belgium)

Tutorial Chairs

John Domingue (Knowledge Media Institute, The Open University)

David Martin (SRI, USA)

Semantic Web in Use Chairs

Dean Allemang (TopQuadrant, USA)

Kyung-Il Lee (Saltlux Inc., Korea)

Lyndon Nixon (Free University Berlin, Germany)

Semantic Web Challenge Chairs

Jennifer Golbeck (University of Maryland, USA)

Peter Mika (Yahoo! Research Barcelona, Spain)

Poster & Demos Chairs

Young-Tack, Park (Sonngsil University, Korea)

Mike Dean (BBN, USA)

Doctoral Consortium Chair

Diana Maynard (University of Sheffield, United Kingdom)

Sponsor Chairs

Young-Sik Jeong (Wonkwang University, Korea)

York Sure (University of Karlsruhe, German)

Exhibition Chairs

Myung-Hwan Koo (Korea Telecom, Korea)

Noboru Shimizu (Keio Research Institute, Japan)

Publicity Chair: Masahiro Hori (Kansai University, Japan)

Proceedings Chair: Philippe Cudré-Mauroux (EPFL, Switzerland

Metadata Chairs

Tom Heath (KMi, OpenUniversity, UK)

Knud Möller (DERI, National University of Ireland, Galway)

EON 2007 Organizing Committee

Raúl García-Castro (Universidad Politécnica de Madrid, Spain)
Denny Vrandecic (AIFB, Universität Karlsruhe (TH), Germany)
Asunción Gómez-Pérez (Universidad Politécnica de Madrid, Spain)
York Sure (EON series inventor), (SAP Research, Germany)
Zhisheng Huang (Vrije University of Amsterdam, The Netherlands)

EON 2007 Program Committee

Harith Alani (University of Southampton, United Kingdom)
Christopher Brewster (University of Sheffield, United Kingdom)
Roberta Cuel (University of Trento, Italy)
Klaas Dellschaft (University of Koblenz, Germany)
Mariano Fernández-López (Universidad San Pablo CEU, Spain)
Jens Hartmann (University of Bremen, Germany)
Kouji Kozaki (Osaka University, Japan)
Joey Lam (University of Aberdeen, United Kingdom)
Thorsten Liebig (Ulm University, Germany)
Enrico Motta (Open University, United Kingdom)
Natasha Noy (Stanford, USA)
Yue Pan (IBM, China)
Elena Paslaru Bontas (DERI Innsbruck, Austria)
Yuzhong Qu (Southeast University, China)
Mari Carmen Suárez-Figueroa (Universidad Politécnica de Madrid, Spain)
Baoshi Yan (Bosch, USA)
Sofia Pinto (INESC-ID, Portugal)

Table of Contents

Characterizing Knowledge on the Semantic Web with WATSON

Mathieu d'Aquin, Claudio Baldassarre, Laurian Gridinoc,
Sofia Angeletou, Marta Sabou, and Enrico Motta*

Knowledge Media Institute (KMi), The Open University, United Kingdom
{m.daquin,c.baldassarre,l.gridinoc,s.angeletou,r.m.sabou,e.motta}@open.ac.uk

Abstract. WATSON is a gateway to the Semantic Web: it collects, analyzes and gives access to ontologies and semantic data available online with the objective of supporting their dynamic exploitation by semantic applications. We report on the analysis of 25 500 ontologies and semantic documents collected by WATSON, giving an account about the way semantic technologies are used to publish knowledge on the Web, about the characteristics of the published knowledge, and about the networked aspects of the Semantic Web. Our main conclusions are 1- that the Semantic Web is characterized by a large number of small, lightweight ontologies and a small number of large-scale, heavyweight ontologies, and 2- that important efforts still need to be spent on improving the published ontologies (coverage of different topic domains, connectedness of the semantic data, etc.) and the tools that produce and manipulate them.

1 Introduction

The vision of a Semantic Web, *"an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation"* [3], is becoming more and more a reality. Technologies like RDF and OWL, allowing to represent ontologies and information in a formal, machine understandable way are now well established. More importantly, the amount of knowledge published on the Semantic Web – i.e, the number of ontologies and semantic documents available online – is rapidly increasing, reaching the critical mass required to enable the vision of a truly large scale, distributed and heterogeneous web of knowledge.

In a previous paper [4], we presented the design and architecture of WATSON, a gateway to the Semantic Web. WATSON is a tool and an infrastructure that automatically collects, analyses and indexes ontologies and semantic data available online in order to provide efficient access to this knowledge for Semantic Web users and applications. Besides enabling the exploitation of the Semantic Web, WATSON can be seen as a research platform supporting the exploration of

* This work was funded by the Open Knowledge and NeOn projects sponsored under EC grant numbers IST-FF6-027253 and IST-FF6-027595

the Semantic Web to better understand its characteristics. This paper reports on the use of this infrastructure to provide quantitative indications about the way semantic technologies are used to publish knowledge on the Web, about the characteristics of the knowledge available online, and about the way ontologies and semantic documents are networked together.

A number of researchers have already produced analyses of the Semantic Web landscape. For example, [6] presents an analysis of 1 300 ontologies looking in particular at the way ontology language primitives are used, and at the distribution of ontologies into the three OWL species (confirming results already obtained in [2]). In [5], the authors of Swoogle present an analysis of the semantic documents collected by Swoogle. The forthcoming section shows complementary results to the ones presented in both these studies, based on a set of almost 25 500 semantic documents collected by WATSON. In particular, in comparison with [5] that focuses on the Web aspects of the Semantic Web (number of files, provenance in terms of website and internet domain, RDF(S) primitive usage, etc.), we consider a more "Semantic Web" centric view, by providing an insight on characteristics like the expressiveness of the employed ontology languages, the structural and domain-related coverage characteristics of semantic documents, and their interconnections in a *knowledge network*.

2 Characterizing Knowledge on the Semantic Web with WATSON

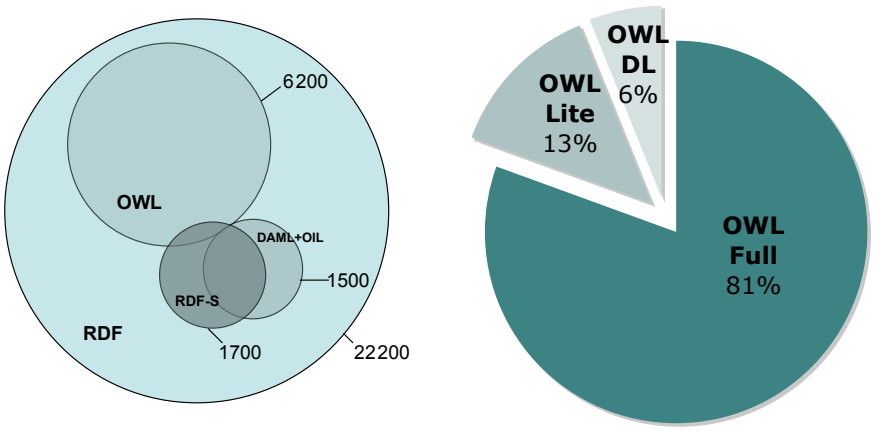
Below, we report on some of the results that have been obtained by collecting, validating and analyzing online ontologies and semantic documents. We focus on three main aspects in this study: the usage of semantic technologies to publish knowledge on the Web (Section 2.1), the characteristics of the knowledge published (Section 2.2) and the connectedness of semantic documents (Section 2.3).

Different sources are used by the WATSON crawler to discover ontologies and semantic data (Google, Swoogle[1], *Ping the Semantic Web.com*[2], etc.) Once located and retrieved, these documents are filtered to keep only valid RDF based documents (by using Jena[3] as a parser). In addition, we have chosen to exclude RSS and FOAF files from the analysis. The main reason to exclude these documents is that RSS and FOAF together represent more than 5 times the number of other RDF documents in our collection. These two vocabularies being dedicated to specific applications, we believe that they would have introduced a bias in our characterization and therefore, that they should be studied separately. We consider here a set of almost 25 500 semantic documents collected by WATSON.

[1] `http://swoogle.umbc.edu/`
[2] `http://pingthesemanticweb.com/`
[3] `http://jena.sourceforge.net/`

2.1 Usage of Semantic Technologies

Semantic technologies such as OWL and RDF are now well established and commonly used by many developers. In this section, we look at the details of how the features provided by Semantic Web languages are exploited to describe ontologies and semantic data on the Web.



(a) (b)

Fig. 1. Usage of the ontology representation languages (a) and of the three OWL species (b).

Representation Languages. Watson implements a simple, but restrictive language detection mechanism. It is restrictive in the sense that it considers a document to employ a particular language only if this document actually *instantiates* an entity of the language vocabulary (any kind of description for RDF, a class for RDF-S, and a class or a property for OWL and DAML+OIL). Figure 1(a) provides a visualization of the results of this language detection mechanism applied on the entire set of semantic documents collected by Watson. A simple conclusion that can be drawn from this diagram is that, while the majority of these documents are exclusively considering factual data in RDF, amongst the ontology representation languages (RDF-S, OWL and DAML+OIL), OWL seems to have been adopted as standard. Another element that is worth to consider is the overlap between these languages. Indeed, our detection mechanism only considers a document to employ two different languages if it actually declares entities in both languages. For example, a document would be considered as being written in both RDF-S and OWL if it contains the definition of an *owl:Class* or an *owl:Property*, together with the definition of an *rdfs:Class*. According to this definition, the use of RDF-S properties like *rdfs:label* is not sufficient to consider the document as being written in RDF-S. Combining entities from two different meta-models, like for example OWL and RDF-S, can

be problematic for the tools that manipulate the ontology (in particular, the inference mechanisms can become undecidable). These considerations have been taken into account in the design of OWL. As a consequence, unlike DAML+OIL documents, most of the OWL documents only employ OWL as an ontology language, leading to cleaner and more exploitable ontologies (see Figure 1(a)).

OWL is divided into three sub-languages, OWL Lite, OWL DL, and OWL Full, that represent different (increasing) levels of complexity. In this respect, the results obtained on the proportion of OWL documents of the three species are surprising (see Figure 1(b)): a large majority of the OWL ontologies are OWL Full. This confirms the results obtained by Wang et al. in [6] on a set of 1 300 ontologies. The explanation provided in [6] is that most ontologies fall into the OWL Full category because of simple syntactic mistakes. This intuition that documents are considered as OWL Full ontologies not because they use the expressive power of this sub-language is confirmed in the next paragraph, which looks at the expressiveness employed by ontologies.

Expressiveness. The Pellet reasoner[4] provides a mechanism to detect the level of expressiveness of the language employed in an ontology in terms of description logics (DLs). DLs are named according to the constructs they provide to describe entities, and so, to their expressive power. For example, the DL of OWL Lite is $\mathcal{ALCR}_{+}\mathcal{HIF}(D)$, meaning for example that it allows the description of inverse relations ($\mathcal{I}$) and of limited cardinality restrictions ($\mathcal{F}$).

Total			**OWL**			**OWL Full**		
DL	*Nb Documents*		*DL*	*Nb Documents*		*DL*	*Nb Documents*	
$\mathcal{AL}$(D)	21375	(84%)	$\mathcal{AL}$(D)	3644	(59%)	$\mathcal{AL}$(D)	3365	(78%)
$\mathcal{AL}$	2455	(10%)	$\mathcal{AL}$	1406	(23%)	$\mathcal{AL}$	281	(6.5%)
$\mathcal{ALH}$(D)	293	(1%)	$\mathcal{ALCF}$(D)	105	(1.5%)	$\mathcal{ALCF}$(D)	68	(1.5%)
$\mathcal{ALCF}$(D)	105	(<1%)	$\mathcal{ALC}$	94	(1.5%)	$\mathcal{ALH}$(D)	44	(1%)
$\mathcal{ALH}$	102	(<1%)	$\mathcal{ALH}$(D)	54	(<1%)	$\mathcal{ALCOF}$(D)	28	(<1%)
$\mathcal{ALC}$	101	(<1%)	$\mathcal{ALCOF}$(D)	43	(<1%)	$\mathcal{ALC}$	27	(<1%)

Table 1. Most common classes of expressiveness employed by semantic documents, on the entire set of semantic documents collected by Watson, on the sub-set of OWL ontologies and on the sub-set of OWL Full ontologies.

Using this mechanism allows us to assess the complexity of semantic documents, i.e., how they employ the expressive power provided by ontology representation languages. Indeed, the analysis presented in Table 1 shows that the advanced features provided by the ontology representation languages are rarely used. $\mathcal{AL}$ is the smallest DL language that can be detected by Pellet. Only adding the use of datatypes (D) and of hierarchies of properties ($\mathcal{H}$) to $\mathcal{AL}$ is sufficient to cover 95% of the semantic documents. It is worth mentioning that these two elements are both features of RDF-S.

[4] `http://www.mindswap.org/2003/pellet/`

Looking at the results for OWL and OWL Full ontologies (second and third parts of Table 1), it appears that the division of OWL in Lite, DL and Full, which is based on the complexity and on the implementation cost, is not reflected in practice. Indeed, the fact that most OWL Full ontologies employ only very simple features confirms the intuition expressed in the previous paragraph: while these ontologies would get the disadvantages of using OWL Full, they do not actually exploit its expressiveness. Moreover, while one of the most popular feature of OWL, the possibility to build enumerated classes ($\mathcal{O}$), is only permitted in OWL DL, transitive and functional properties ($\mathcal{R}+$), which are features of OWL Lite, are rarely used.[5]

2.2 Structural and Topic Coverage Characteristics of Knowledge on the Semantic Web

One important aspect to consider for the exploitation of the Semantic Web concerns the characteristics of the semantic documents in terms of structure and topic coverage. In this section, we report on the analysis of these aspects from the data provided by the WATSON repository with the objective of helping users and developers in knowing what they can expect from the current state of the Semantic Web.

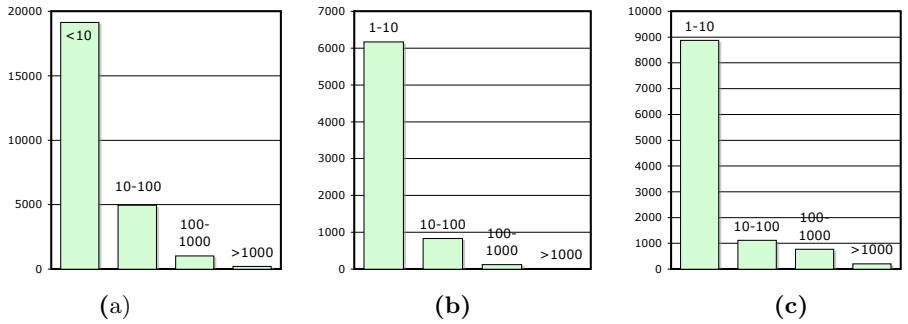


Fig. 2. Number of semantic documents (y axis) in 4 categories of size, in terms of the total number of entities (a), classes (b), and individuals(c).

Size. As already mentioned, WATSON has collected almost 25 500 distinct semantic documents (by distinct we mean that if the same file appears several times, it is counted only once, see Section 2.3). Within these documents, about 1.1 million distinct entities (i.e. classes, properties, and individuals having different URIs) have been extracted.

[5] Considering only features not handled by RDF-S (i.e. excluding $\mathcal{ALH}(\mathcal{D})$), $\mathcal{O}$ is the third most used feature of OWL with 236 ontologies, after $\mathcal{C}$ (748) and $\mathcal{F}$ (598), while $\mathcal{R}+$ is last with only 31 ontologies.

An interesting information that can be extracted from this analysis is that ontologies on the Semantic Web are generally of very small size. Indeed, the average number of entities in semantic documents is around 43, that is far closer to the minimum size of semantic documents (1 entity) than to the bigger one (more than 28 000 entities). Looking more in detail, it can be seen that the Semantic Web is in fact characterized by a large number of very small documents, and a small number of very large ones (see Figure 2(a)). It is worth mentioning that, as shown in Figures 2(b) and 2(c), this observation is valid for both ontological knowledge and factual data.

Measures	**Value**
Total number of classes	161 264
Total number of properties	76 350
Total number of individuals	984 526
Total number of domain relations	32 572
Total number of sub-class relations	106 729
Total number of instance relations	1 114 795
average **P-density** (number of properties per class)	**0.20**
average **H-density** (number of super-classes per class)	**0.66**
average **I-density** (number of instances per class)	**6.9**

Table 2. Measures of density over the WATSON repository.

Density. One way to estimate the richness of the representation in semantic documents is to rely on the notion of density. Extending the definition provided by [1], we consider the density of a semantic entity to be related to its interconnection with other entities. Accordingly, different notions of density are considered: the number of properties for each class (P-density), the number of super-classes for each class (H-density), and the number of instances for each class (I-density). In the case of P-Density, a class is considered to possess a property if it is declared as the domain of this property. It is worth mentioning that none of these measures takes inheritance into consideration: only directly stated relations are counted. Computing these measures on the whole WATSON repository (see Table 2) allows us to conclude that, on average, ontology classes are described in a lightweight way (this correlates with the results obtained in the previous section concerning the expressiveness of the employed language). More precisely, the P-density and H-density measures tend to be low on average, in particular if compared to their maximum (17 and 47 respectively). Moreover, it is often the case that ontologies would contain a few "central", richly described classes. This characteristic cannot be captured by simply looking at the average density of the collected entities. Therefore, we looked at the maximum density within one ontology (i.e. the density of the densest class in the ontology). The *average maximum P-density in ontologies that contain domain relations* is still low (1.1), meaning that, in most cases, classes may at most possess only 1 property, if any. Similar results are obtained for H-density (1.2 average maximum H-density in ontologies having sub-class relations).

Another straightforward conclusion here is that the amount of instance data is much bigger than the amount of ontological knowledge in the collected semantic documents. It is expected that the Semantic Web as a whole would be built on a similar ratio of classes, properties and individuals, requiring ontology based tools to handle large repositories of instances.

Topic Coverage. Understanding the topic coverage of the Semantic Web, i.e. how ontologies and semantic documents relate to generic topic domains like health or business, is of particular importance for the development of semantic applications. Indeed, even if it has already been demonstrated that the Semantic Web is rapidly growing [5], we cannot assume that this increase of the amount of online knowledge has been achieved in the same way for every application domain.

The WATSON analysis task includes a mechanism that categorizes ontologies into the 16 top groups of DMOZ[6]. Each category is described by a set of weighted terms, corresponding to the name of its sub-categories in DMOZ. The weight $w(t) = \frac{1}{l(t)} \times \frac{1}{f(t)}$ of a term t is calculated using the level $l(t)$ of the corresponding sub-category in DMOZ and the number of times $f(t)$ the term is used as a sub-category name. In this way, a term would be considered as a good descriptor for the category (has a high weight) if it is high in the corresponding sub-hierarchy and if is is rarely used to describe other categories. The *level of coverage* of a given ontology to a given category then corresponds to the sum of the weight of the terms that match (using a simple lexical comparison) entities in the ontology.

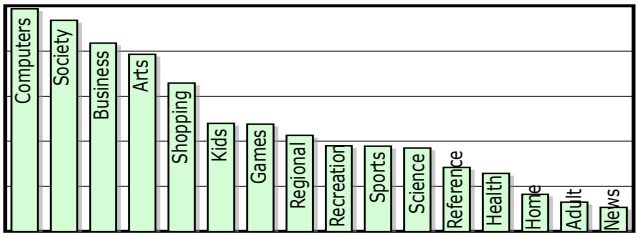


Fig. 3. Relative coverage of the 16 topics corresponding to the top categories of the DMOZ topic hierarchy.

This simple mechanism allows us to compute a rough overview of the relative coverage of these 16 high level topics on the Semantic Web. Among the semantic documents collected by WATSON, almost 7 000 have been associated to one or several topics (have a non null level of coverage on some topics). Figure 3 describes the relative coverage of the 16 considered topics. In this figure, the y axis corresponds to the sum of the levels of coverage of all ontologies for the considered topic. The actual numbers here are not particularly significant, as we

[6] `http://dmoz.org/`

are more interested in the differences in the level of coverage for different topics. As expected, it can be seen that, while some topics are relatively well covered (e.g. computers, society, business), others are almost absent from the collected semantic documents (home, adult, news). Also, when comparing these results to the distribution of web documents within the DMOZ hierarchy, it is interesting to find that, according to this categorization, the coverage of these topics on the "classical Web" is also rather unbalanced (with categories varying from 31 294 to 1 107 135 documents), but that the order of the topics according to coverage is very different (computers for example is the 6^{th} category in coverage).

Finally, by looking at the level of coverage of each ontology, the *power law* distribution that has been found for other characteristics (size, expressiveness) also applies here: a few semantic documents have a high level of coverage, often with respect to several topics, whereas the large majority have a very low level of coverage, with respect to one or two topics only.

2.3 The Knowledge Network

While the Web can be seen as a network of documents connected by hyperlinks, the Semantic Web is a network of ontologies and semantic data. This aspect also needs to be analyzed, looking at the semantic relations linking semantic documents together.

Connectedness. Semantic documents and ontologies are connected through references to their respective namespaces. While the average number of references to external namespaces in the documents collected by Watson seems surprisingly high (6.5), it is interesting to see that the most referenced namespaces are very often hosted under the same few domains (`w3.org`, `stanford.edu`, `ontoworld.org`, etc.)[7] This seems to indicate that a small number of large, dense "nodes" tend to provide the major part of the knowledge that is reused.

Another element of importance when considering the inter-connection between online semantic data is whether the URIs used to describe entities are *dereferenceable*, i.e., wether the namespaces to which they belong correspond to an actual location (a reachable URL) from which descriptions of the entities can be retrieved. Several applications, like Tabulator[8] or the *Semantic Web Client Library*[9] are indeed based on this assumption: that the Semantic Web can be *traversed* through dereferenceable URIs. However, among the semantic documents that explicitly declare their namespace, only about 30% correspond to actual locations of semantic documents, which means that these applications can only access a restricted part of the Semantic Web.

Redundancy. As in any large-scale distributed environment, redundancy is inevitable on the Semantic Web and actually contributes to its robustness: it

[7] It is important to remark here that the references to the namespaces of the representation languages, such as RDF and OWL, were not counted.

[8] `http://www.w3.org/2005/ajar/tab`

[9] `http://sites.wiwiss.fu-berlin.de/suhl/bizer/ng4j/semwebclient/`

is useful for an application to know that the semantic resources it uses can be retrieved from alternative locations in case the one it relies on becomes unreachable. As already mentioned, the 25 500 documents collected by WATSON are distinct, meaning that if the same file is discovered several times, it is only stored and analyzed once, even if WATSON would keep track of all its locations. On average, every semantic document collected by WATSON can be found in 1.27 locations, meaning that around 32 350 URLs actually address semantic data or ontologies. Ingnoring this simple phenomenon, like it is the case for example with the analysis described in [5], would have introduced an important bias in our analysis.

At a more fine-grained level, descriptions of entities can also be distributed and do not necessarily exist in a single file. Pieces of information about the same entity, identified by its URI, can be physically declared at different locations. Indeed, among the entities collected by WATSON, about 12% (approximately 150 000) are described in more than one place.

URI duplication. In theory, if two documents are identified by the same URI, they are supposed to contribute to the same ontology, i.e. the entities declared in these documents are intended to belong to the same conceptual model. This criterion is consistent with the distributed nature of the Semantic Web in the sense that ontologies can be physically distributed among several files, on different servers. However, even if this situation appears rarely (only 60 URIs of documents are "non unique"), in most cases, semantic documents that are identified by the same URI are not intended to be considered together. We can distinguish different situations leading to this problem:

Default URI of the ontology editor. `http://a.com/ontology` is the URI of 20 documents that do not seem to have any relation with each other, and that are certainly not meant to be considered together in the same ontology. The reason for this URI to be so popular is that it is the default namespace attributed to ontologies edited using (some of the versions) of the OWL Plugin of the Protégé editor[10]. Systematically asking the ontology developer to give an identifier to the edited ontology, like it is done for example in the SWOOP editor[11], could avoid this problem.

Mistaken use of well known namespaces. The second most commonly shared URI in the WATSON repository is `http://www.w3.org/2002/07/owl`, which is the URI of the OWL schema. The namespaces of RDF, RDF Schema, and of other well known vocabularies are also often duplicated. Using these namespaces as URIs for ontologies is (in most cases) a mistake that could be avoided by checking, prior to giving an identifier to an ontology, if this identifier has already been used in another ontology.

Different versions of the same ontology. A third common reason for which different semantic documents share the same URI is in situations where an ontology evolves to a new version, keeping the same URI (e.g., `http://lsdis.cs.uga.edu/proj/semdis/testbed/`). As it is the same ontology, it

[10] `http://protege.stanford.edu/`
[11] `http://www.mindswap.org/2004/SWOOP/`

seems natural to keep the same URI, but in practice, this can cause problems in these cases where different versions co-exist and are used at the same time. This leads to a need for recommendations of good practices on the identification of ontologies, that would take into account the evolution of the ontologies, while keeping different versions clearly separated.

3 Conclusion

The main motivation behind WATSON is that the Semantic Web requires efficient infrastructures and access mechanisms to support the development of a new kind of applications, able to exploit dynamically the knowledge available online. We believe that a better understanding of the current practices concerning the fundamental characteristics of the Semantic Web is required. In this paper, we have reported on the analysis of the 25 500 distinct semantic documents collected by WATSON, giving an account about the way semantic technologies are used to publish knowledge on the Web, about the characteristics of the published knowledge, and about some of the networked aspects of the Semantic Web. Our main conclusions are 1- that the Semantic Web is characterized by a large number of small, lightweight ontologies and a small number of large-scale, large-coverage and heavyweight ontologies, and 2- that important efforts still need to be spent on improving published ontologies (coverage of different domains, connectedness of the semantic data, etc.) and the tools that produce and manipulate them.

Many other aspects and elements could have been analyzed, and the research work presented here can be seen as a first step towards a more complete characterization of the Semantic Web. In particular, we only considered the characterization of the *current* state of the Semantic Web, analyzing a *snapshot* of the online semantic documents that represent the WATSON repository. In the future, we plan to also consider the dynamics of the Semantic Web, looking at how the considered characteristics evolve over time.

References

1. H. Alani, C. Brewster, and N. Shadbolt. Ranking Ontologies with AKTiveRank. In *Proc. of the International Semantic Web Conference, ISWC*, 2006.
2. S. Bechhofer and R. Volz. Patching Syntax in OWL ontologies. In *Proc. of International Semantic Web Conference, ISWC*, 2004.
3. T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, May 2001.
4. M. d'Aquin, M. Sabou, M. Dzbor, C. Baldassarre, L. Gridinoc, S. Angeletou, and E. Motta. WATSON: A Gateway for the Semantic Web. In *Proc. of European Semantic Web Conference, ESWC, Poster Session*, 2007.
5. L. Ding and T. Finin. Characterizing the Semantic Web on the Web. In *Proc. of International Semantic Web Conference, ISWC*, 2006.
6. T. D. Wang, B. Parsia, and J. Hendler. A Survey of the Web Ontology Landscape. In *Proc. of the International Semantic Web Conference, ISWC*, 2006.

Evaluating Ontology Search

Paul Buitelaar, Thomas Eigner

German Research Center for Artificial Intelligence (DFKI GmbH)
Language Technology Lab & Competence Center Semantic Web
Stuhlsatzenhausweg 3
Saarbrücken, Germany
paulb@dfki.de

Abstract. As more and more ontologies are being published on the Semantic Web, selecting the most appropriate ontology will become an increasingly important subtask in Semantic Web applications. Here we present an approach towards ontology search in the context of OntoSelect, a dynamic web-based ontology library. In OntoSelect, ontologies can be searched by keyword or by document. In keyword-based search only the keyword(s) provided by the user will be used for the search. In document-based search the user can provide either a URL for a web document that represents a specific topic or the user simply provides a keyword as the topic which is then automatically linked to a corresponding Wikipedia page from which a linguistically/statistically derived set of most relevant keywords will be extracted and used for the search. In this paper we describe an experiment in evaluating the document-based ontology search strategy based on an evaluation data set that we constructed specifically for this task.

1 Introduction

A central task in the Semantic Web effort is the semantic annotation or knowledge markup of data (textual or multimedia documents, structured data, etc.) with semantic metadata as defined by one or more ontologies. The added semantic metadata allow for automatic processes (agents, web services, etc.) to interpret the underlying data in a unique and formally specified way, thereby enabling autonomous information processing. As ontology-based semantic metadata are in fact class descriptions, the annotated data can be extracted as instances for these classes. Hence, another way of looking at ontology-based semantic annotation is as ontology population.

Most of current work in ontology-based semantic annotation assumes ontologies that are typically developed specifically for the task at hand. Instead, a more realistic approach would be to access an ontology library and to select one or more appropriate ontologies. Although the large-scale development and publishing of ontologies is still only in a beginning phase, many are already available. To select the most appropriate ontology (or a combination of complementary ontologies) will therefore be an increasingly important subtask of Semantic Web applications.

Until very recently the solution to this problem was supposed to be handled by foundational ontology libraries [1,2]. However, in recent years, dynamic web-based ontology libraries and ontology search engines like OntoKhoj [3], OntoSelect [4],

SWOOGLE [5] and Watson [6] have been developed that enable a more data-driven approach to ontology search and retrieval.

In OntoSelect, ontologies can be searched by keyword or by document. In keyword-based search only the keyword(s) provided by the user will be used for the search. In document-based search the user can provide either a URL for a web document that represents a specific topic or the user simply provides a keyword as the topic which is then automatically linked to a corresponding Wikipedia page from which a linguistically/statistically derived set of most relevant keywords will be extracted and used for the search. In this paper we describe an experiment in evaluating the document-based ontology search strategy based on an evaluation data set that we constructed specifically for this task.

The remainder of the paper is structured as follows. Section 2 gives a brief overview of the content and functionality of the OntoSelect ontology library. Section 3 presents a detailed overview of the ontology search algorithm and scoring method used. Section 4 presents the evaluation benchmark, experiments and results. Finally, section 5 presents some conclusions and gives an outlook on future work

2 The OntoSelect Ontology Library

OntoSelect is a dynamic web-based ontology library that collects, analyzes and organizes ontologies published on the Semantic Web. OntoSelect allows browsing of ontologies according to size (number of classes, properties), representation format (DAML, RDFS, OWL), connectedness (score over the number of included and referring ontologies) and human languages used for class- and object property-labels. OntoSelect further includes an ontology search functionality as described above and discussed in more detail in the following sections.

OntoSelect uses the Google API to find published ontologies on the web in the following formats: DAML, OWL and RDFS. Jena is used for reading and analyzing the ontologies. In the case of OWL, OntoSelect also determines its type (Full, DL, Lite) and indexes this information accordingly. Each class and object property defined by the ontology is indexed with reference to the ontology in which it occurs. Correspondingly, each label is indexed with reference to the corresponding ontology, class or object property, the human language of the label (if available), and a normalized label name, e.g. TaxiDriver is normalized to “taxi driver”. Object properties are handled similarly as classes except that also information on their type (functional, transitive, symmetric) is indexed. Finally, a separate index is build up in which we keep track of the distribution of labels over all of the collected ontologies. In this way, a ranked list of frequently used labels can be maintained and browsed by the user.

3 Ontology Search

3.1 Ontology Search Measures and Criteria

The ontology search problem is a very recent topic of research, which only originated with the growing availability of ontologies on the web. A web-based ontology, defined

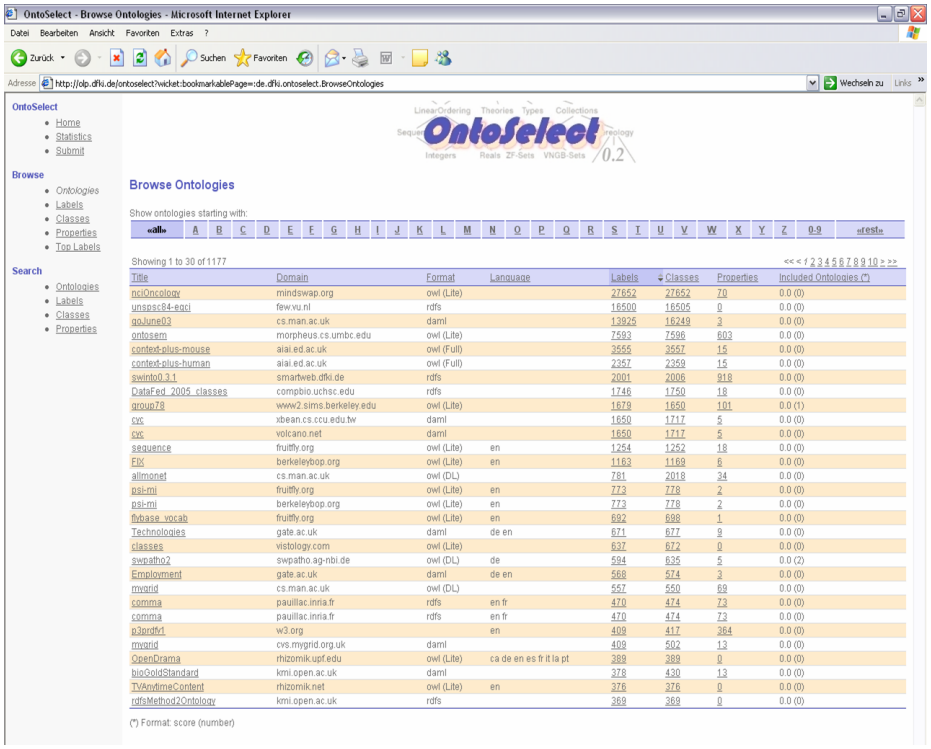


Fig. 1. Browsing ontologies in OntoSelect

by representation languages such as OWL or RDFS, is in many respects just another web document that can be indexed, stored and retrieved. On the other hand, an ontology is a highly structured document with possibly explicit semantic links to other ontologies. The OntoSelect approach is based on both observations by ranking ontologies by coverage, i.e. the overlap between query terms and index terms; by structure, i.e. the ratio of class vs. property definitions; and by connectedness, i.e. the level of integration between ontologies.

Other approaches have similarly stressed the importance of such measures, e.g. [7] describe the “Class Match”, “Density”, “Semantic Similarity” and “Betweenness” measures. The Class Match and Density measures correspond roughly to our coverage and structure measure, whereas the Semantic Similarity and Betweenness measure the semantic weight of query terms relative to the different ontologies that are to be ranked. These last two measures are based on the assumption that ontologies are well-structured with equal semantic balance throughout all constitutive parts, which unfortunately is only seldom the case and we therefore do not take such measures into account.

Another set of measures or rather criteria for ontology search has been proposed by [8]. The focus here is more on the application of found ontologies and therefore includes such criteria as: ‘modularization’ (can retrieved ontologies be split up in useful

modules); 'returning ontology combinations' (can retrieved ontologies be used in combination); 'dealing with instances' (do retrieved ontologies include instances as well as classes/properties).

These criteria are desirable but are currently not central to the OntoSelect approach and to this paper. Our focus is rather on providing data-driven methods for finding the best matching ontology for a given topic and on providing a proper evaluation of these methods.

3.2 Ontology Search in OntoSelect

Ontology ranking in OntoSelect is based on a combined measure of *coverage*, *structure* and *connectedness* of ontologies as discussed above. Further, OntoSelect provides automatic support in ontology ranking relative to a web document instead of just one or more keyword(s). Obviously this allows for a much more fine-grained ontology search process.

For a given document as search query, OntoSelect first extracts all textual data and analyses this with linguistic tools (i.e. 'part-of-speech tagger' and 'morphological analysis') to extract and normalize all nouns in the text as these can be expected to represent ontology classes rather than verbs, adjectives, etc. The frequencies of these nouns in the query document is then compared with their frequencies in a reference corpus - consisting of a large collection of text documents on many different topics and covering a large section of the English language - to estimate the relevance for each noun based on how often it is expected to appear in a more general text of the same size. Chi-square is used to estimate this relevance score (see also Coverage score below). Only the top 20 nouns are used further in the search process as extracted keywords.

To calculate the relevance of available ontologies in OntoSelect, the set of 20 extracted keywords is used to compute three separate scores (*coverage*, *structure*, *connectedness*) and a combined score as described below:

Coverage: How many of the terms in the document are covered by the labels in the ontology?

To estimate the coverage score, OntoSelect iterates over all ontologies containing at least one label (either the original label name or the normalized label name) occurring in the top 20 keyword list of the search document. For each label occurring in the document, OntoSelect computes its relevance, with which the coverage score of an ontology O is calculated.

$$
\begin{aligned}
QD &= \text{Query Document} \\
KW &= \text{Set of extracted keywords of } QD \\
OL &= \text{Set of labels for ontology } O \\
RefC &= \text{Reference Corpus} \\
Exp_k &= \frac{RefC_k}{|RefC|} \times |QD| \\
\chi^2(k) &= \frac{QD_k - Exp_k}{Exp_k} \\
coverage(O, QD) &= \sum_{k \in KW(QD) \cap OL(O)} \chi^2(k)
\end{aligned}
$$

Connectedness: Is the ontology connected to other ontologies and how well established are these?
Similar to the Google PageRank algorithm [9], OntoSelect checks how many ontologies import a specific ontology, but also how many ontologies are imported by that one. The connectedness score of an ontology O is calculated accordingly.

$$
\begin{aligned}
cIO(O) &= \text{number of imported Ontologies for } O \\
cIRO(O) &= \text{number of imported Ontologies} \\
&\quad \text{(that could be parsed) for } O \\
cIFO(O) &= \text{number of Ontologies importing } O \\
IO(O) &= \{\, x \mid x \text{ imports the Ontology } O\} \\
iS(O, level) &= \tfrac{cIFO(O)}{2^{level}} + \textstyle\sum_{O' \in IO(O)} iS(O', level+1) \\
connectedness(O) &= \begin{cases} cIO(O) > 0 : \frac{iS(O,0) * cIO(O)}{iS(O,0) cIO(O)} \times \frac{countIRO(O)}{countIO(O)} \\ else : \qquad\qquad 0 \end{cases}
\end{aligned}
$$

Structure: How detailed is the knowledge structure that the ontology represents?
Structure is measured by the number of properties relative to the number of classes of the Ontology O. This parameter is based on the observation that more advanced ontologies generally have a large number of properties. Therefore, a relatively large number of properties would indicate a highly structured and hence more advanced ontology.

$$structure(O) = \frac{\text{\# of properties in ontology O}}{\text{\# of classes in ontology O}}$$

Combined Score: Since the ranges of coverage, connectedness and structure are very discrepant these values have to be normalized. In other words, all coverage values are divided by the maximum coverage value, all connectedness values by the maximum connectedness value and all structure values by the maximum structure value, giving rise to final values between 0 and 1. Because each type of score has a different significance, the final score is a weighted combination of the three individual score.

$$score = \frac{3 \times coverage_{norm} + 2 \times connectedness_{norm} + structure_{norm}}{6}$$

3.3 An Example of Ontology Search in OntoSelect

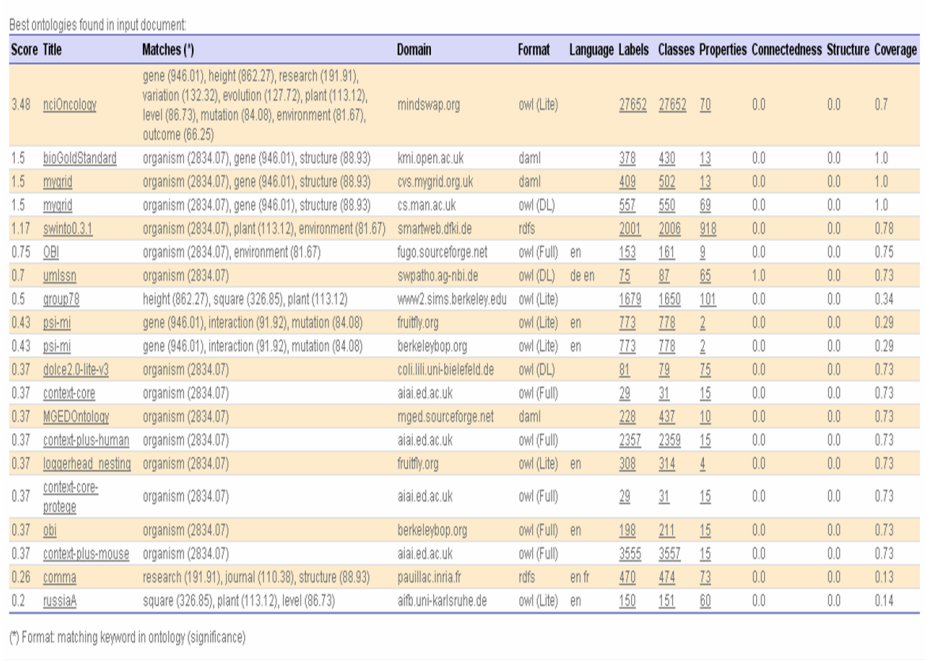
The application of the ranking and search algorithm discussed above can be illustrated with an example of ontology search on the topic 'genetics', which may be represented by the Wikipedia page on 'Gene':

http://en.wikipedia.org/wiki/Gene

The results of the keyword extraction and ontology ranking process for this query document are reported by OntoSelect in two tables, one that shows the top 20 keywords extracted from the query document and one with the ranked list of best matching ontologies according to the computed score (see Figure 2). Combined and individual

scores - connectedness, structure, coverage - are shown as well as the matching labels/keywords and their relevance scores. Extracted and top ranked keywords include "gene", "molecule", "transcription", "protein", etc., all of which are indeed of relevance to the 'genetics' topic.

Retrieved and top ranked ontologies include a large number that are indeed of relevance to the 'genetics' topic, e.g. "nciOncology", "bioGoldStandard", "mygrid", "sequence", etc. Only some of the ontologies are not or less relevant, e.g. "swinto" (which is mainly on football but also includes all of SUMO that does in fact cover many terms that are relevant to genetics), "gold" (which is mainly on linguistics but includes some terms that have also some relevance to genetics), "dolce" (which is a foundational top ontology that includes some terms with relevance to genetics).



Best ontologies found in input document:

Score	Title	Matches (*)	Domain	Format	Language	Labels	Classes	Properties	Connectedness	Structure	Coverage
3.48	nciOncology	gene (946.01), height (862.27), research (191.91), variation (132.32), evolution (127.72), plant (113.12), level (86.73), mutation (84.08), environment (81.67), outcome (66.25)	mindswap.org	owl (Lite)		27652	27652	70	0.0	0.0	0.7
1.5	bioGoldStandard	organism (2834.07), gene (946.01), structure (88.93)	kmi.open.ac.uk	daml		378	430	13	0.0	0.0	1.0
1.5	mygrid	organism (2834.07), gene (946.01), structure (88.93)	cvs.mygrid.org.uk	daml		409	502	13	0.0	0.0	1.0
1.5	mygrid	organism (2834.07), gene (946.01), structure (88.93)	cs.man.ac.uk	owl (DL)		557	550	69	0.0	0.0	1.0
1.17	swinto0.3.1	organism (2834.07), plant (113.12), environment (81.67)	smartweb.dfki.de	rdfs		2001	2006	918	0.0	0.0	0.78
0.75	OBI	organism (2834.07), environment (81.67)	fugo.sourceforge.net	owl (Full)	en	153	161	9	0.0	0.0	0.75
0.7	umlssn	organism (2834.07)	swpatho.ag-nbi.de	owl (DL)	de en	75	87	65	1.0	0.0	0.73
0.5	group78	height (862.27), square (326.85), plant (113.12)	www2.sims.berkeley.edu	owl (Lite)		1679	1650	101	0.0	0.0	0.34
0.43	psi-mi	gene (946.01), interaction (91.92), mutation (84.08)	fruitfly.org	owl (Lite)	en	773	778	2	0.0	0.0	0.29
0.43	psi-mi	gene (946.01), interaction (91.92), mutation (84.08)	berkeleybop.org	owl (Lite)	en	773	778	2	0.0	0.0	0.29
0.37	dolce2.0-lite-v3	organism (2834.07)	coli.lili.uni-bielefeld.de	owl (DL)		81	79	75	0.0	0.0	0.73
0.37	context-core	organism (2834.07)	aiai.ed.ac.uk	owl (Full)		29	31	15	0.0	0.0	0.73
0.37	MGEDOntology	organism (2834.07)	mged.sourceforge.net	daml		228	437	10	0.0	0.0	0.73
0.37	context-plus-human	organism (2834.07)	aiai.ed.ac.uk	owl (Full)		2357	2359	15	0.0	0.0	0.73
0.37	loggerhead_nesting	organism (2834.07)	fruitfly.org	owl (Lite)	en	308	314	4	0.0	0.0	0.73
0.37	context-core-protege	organism (2834.07)	aiai.ed.ac.uk	owl (Full)		29	31	15	0.0	0.0	0.73
0.37	obi	organism (2834.07)	berkeleybop.org	owl (Full)	en	198	211	15	0.0	0.0	0.73
0.37	context-plus-mouse	organism (2834.07)	aiai.ed.ac.uk	owl (Full)		3555	3557	15	0.0	0.0	0.73
0.26	comma	research (191.91), journal (110.38), structure (88.93)	pauillac.inria.fr	rdfs	en fr	470	474	73	0.0	0.0	0.13
0.2	russiaA	square (326.85), plant (113.12), level (86.73)	aifb.uni-karlsruhe.de	owl (Lite)	en	150	151	60	0.0	0.0	0.14

(*) Format: matching keyword in ontology (significance)

Fig. 2. Ranked list of retrieved ontologies for Wikipedia page 'Gene'

4 Evaluation

In order to test the accuracy of our approach we designed an evaluation experiment with a specifically constructed benchmark of 57 ontologies from the OntoSelect library that were manually assigned to 15 different topics, each of which represented by one or more Wikipedia pages. In this way we were able to define ontology search as a regular information retrieval task, for which we can give relevance assessments (manual

assignment of ontology documents to Wikipedia-based topics) and compute precision and recall for a set of queries (Wikipedia pages). In the following we describe the evaluation benchmark in some more detail as well as the evaluation process and results.

4.1 Evaluation Benchmark

The evaluation experiment is based on a benchmark that consists of 15 Wikipedia topics and 57 out of 1056 ontologies that have been collected through OntoSelect. The 15 Wikipedia topics covered by the evaluation benchmark were selected out of the set of all class/property labels in OntoSelect - 37284 in total - by the following steps:

- Filtering out labels that did not correspond to a Wikipedia page - this left us with 5658 labels (i.e. topic candidates)
- Next, the 5658 labels were used as search terms in SWOOGLE to filter out labels that returned less than 10 ontologies (out of the 1056 in OntoSelect) - this left us with 3084 labels / topics
- We then manually decided which of these 3084 labels actually expressed a useful topic, e.g. we left out very short labels ('v') and very abstract ones ('thing') - this left us with 50 topics
- Finally, out of these 50 we randomly selected 15 for which we manually checked the ontologies retrieved from OntoSelect and SWOOGLE - in this step we checked 269 ontologies out of which 57 were judged as appropriate for the corresponding topic

The resulting 15 Wikipedia topics with the number of appropriately assigned ontologies are: Atmosphere (2), Biology (11), City (3), Communication (10), Economy (1), Infrastructure (2), Institution (1), Math (3), Military (5), Newspaper (2), Oil (0), Production (1), Publication (6), Railroad (1), Tourism (9) For instance, the following 3 ontologies could be assigned to the topic (Wikipedia page) City:

- http://www.mindswap.org/2003/owl/geo/geoFeatures.owl
- http://www.glue.umd.edu/ katyn/CMSC828y/location.daml
- http://www.daml.org/2001/02/geofile/geofile-ont

4.2 Experiment and Results

Based on the evaluation benchmark we defined an experiment that measures how accurate the OntoSelect ontology ranking and search algorithm returns results for each of the topics in the benchmark and compare results with SWOOGLE. Average precision for OntoSelect and SWOOGLE is shown in Figure 3 with detailed results presented in Table 1. The first two columns present the benchmark, against which the experiment is evaluated. The third and fourth columns show recall, precision and F-measure computed over the top 20 retrieved ontologies in OntoSelect and SWOOGLE respectively.

Results unfortunately show that OntoSelect on average performs worse than SWOOGLE, although for selected topics OntoSelect does give better results. In current work we are therefore improving our search algorithm in various ways, e.g. by introducing a centrality score for individual classes - and therefore also for corresponding labels that are to be matched with the search topic and related keywords.

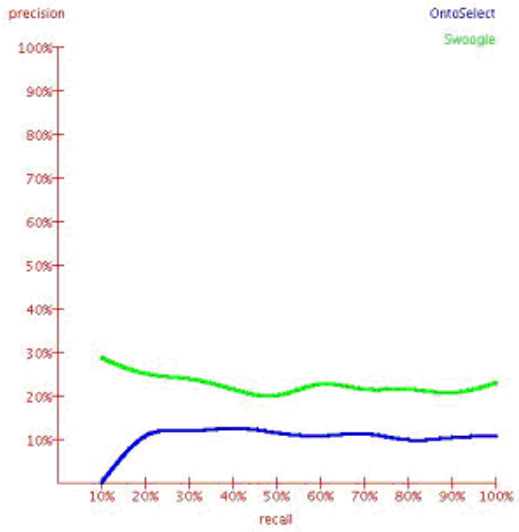


Fig. 3. Average precision for OntoSelect and SWOOGLE

Benchmark		OntoSelect			SWOOGLE		
Topic	**Assigned Ontologies**	**Rec.**	**Prec.**	**F**	**Rec.**	**Prec.**	**F**
Atmosphere	2	0.5	0.1	0.2	1.0	0.2	0.3
Biology	11	0.7	0.8	0.8	0.1	0.1	0.1
City	3	0.3	0.1	0.2	0.3	0.1	0.2
Communication	10	0	0	0	0.6	0.6	0.6
Economy	1	0	0	0	1.0	0.1	0.2
Infrastructure	2	0.5	0.1	0.2	1.0	0.2	0.3
Institution	1	0	0	0	0	0	0
Math	3	0.3	0.1	0.2	1.0	0.3	0.5
Military	5	0	0	0	0.6	0.3	0.4
Newspaper	2	0.5	0.1	0.2	0.5	0.1	0.2
Oil	0	0	0	0	0	0	0
Production	1	1.0	0.1	0.2	0	0	0
Publication	6	0.2	0.1	0.1	0.3	0.2	0.3
Railroad	1	0	0	0	1.0	0.1	0.2
Tourism	9	0	0	0	1.0	0.9	0.9

Table 1. Detailed results over all 15 topics

More in general however, we see our contribution in establishing an evaluation benchmark for ontology search that will enable us to improve the OntoSelect search service in a systematic way. As we intend to make this evaluation benchmark (the 'OntoSelect data set') publicly available, we hope this will also be of use to the Semantic Web community and will allow for better comparison between different systems and methods.

5 Conclusions and Future work

We discussed the OntoSelect search algorithm and described an experiment in evaluating this against an evaluation benchmark (the 'OntoSelect data set') that we constructed specifically for this task. The benchmark consists of 15 topics (represented by Wikipedia pages) that were manually assigned to 57 ontologies from a set of 1056 that were collected automatically through OntoSelect. The evaluation experiment has shown that OntoSelect on average performs worse than SWOOGLE, although for selected topics OntoSelect does give better results. In future work we will further investigate the reasons for this, e.g. we currently investigate the influence of centrality of classes relative to an ontology which may be used to reduce the relevance of general ontologies such as SUMO (as included in the SWIntO ontology). We also intend to extend the evaluation benchmark towards 50 topics and make this resource publicly available.

Demonstration

The OntoSelect ontology library and ontology search is available at:

http://olp.dfki.de/OntoSelect/

Acknowledgements

We thank Michael Velten for implementing the current version of OntoSelect and Bogdan Sacaleanu for providing us with useful comments and insights on the evaluation experiments. This research has been supported in part by the SmartWeb project, which is funded by the German Ministry of Education and Research under grant 01 IMD01.

References

1. G. van Heijst, A.T. Schreiber, and B.J. Wielinga. Using explicit ontologies in KBS development. *International Journal of Human-Computer Studies*, 46(2/3):183–292, 1997.
2. Y. Ding and D. Fensel. Ontology Library Systems: The key to successful Ontology Re-use. *Proceedings of the First Semantic Web Working Symposium. California, USA: Stanford University*, pages 93–112, 2001.
3. C. Patel, K. Supekar, Y. Lee, and EK Park. OntoKhoj: a semantic web portal for ontology searching, ranking and classification. *Proceedings of the fifth ACM international workshop on Web information and data management*, pages 58–61, 2003.

4. P. Buitelaar, T. Eigner, and T. Declerck. OntoSelect: A Dynamic Ontology Library with Support for Ontology Selection. *Proceedings of the Demo Session at the International Semantic Web Conference. Hiroshima, Japan*, 2004.
5. L. Ding, T. Finin, A. Joshi, R. Pan, R.S. Cost, Y. Peng, P. Reddivari, V. Doshi, and J. Sachs. Swoogle: a search and metadata engine for the semantic web. *Proceedings of the Thirteenth ACM conference on Information and knowledge management*, pages 652–659, 2004.
6. M. d'Aquin, M. Sabou, M. Dzbor, C. Baldassarre, S. Gridinoc, L. Angeletou, and Motta E. WATSON: A Gateway for the Semantic Web. *In Proceedings of the 5th International Semantic Web Conference (ISWC), Georgia, USA*, 2005.
7. H. Alani, C. Brewster, and N. Shadbolt. Ranking Ontologies with AKTiveRank. *Poster session of the European Semantic Web Conference, ESWC*, 2006.
8. M. Sabou, V. Lopez, E. Motta, and V. Uren. Ontology Selection: Ontology Evaluation on the Real Semantic Web. *Proceedings of the Evaluation of Ontologies on the Web Workshop, held in conjunction with WWW*, 2006, 2006.
9. L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web, 1998.

Benchmarking Reasoners for Multi-Ontology Applications

Ameet N Chitnis, Abir Qasem and Jeff Heflin

Lehigh University, 19 Memorial Drive West, Bethlehem, PA 18015
{anc306, abq2, heflin}@cse.lehigh.edu

Abstract. We describe an approach to create a synthetic workload for large scale extensional query answering experiments. The workload comprises multiple interrelated domain ontologies, data sources which commit to these ontologies, synthetic queries and map ontologies that specify a graph over the domain ontologies. Some of the important parameters of the system are the average number of classes and properties of the source ontology which are mapped with the terms of target ontology and the number of data sources per ontology. The ontology graph is described by various parameters like its diameter, number of ontologies and average out-degree of node ontology. These parameters give a significant degree of control over the graph topology. This graph of ontologies is the central component of our synthetic workload that effectively represents a web of data.

1 Introduction

One of the primary goals of the Semantic Web is to be able to integrate data from diverse sources irrespective of the ontology to which it commits to. Unfortunately it is difficult to measure progress against this goal. Although there are a large number of ontologies, few have data associated with them, thereby making it difficult to execute large scale integration experiments. The aim of this paper is to provide a benchmark for a synthetic workload that can be easily scaled to the desired configuration for executing large scale extensional query answering experiments.

The benchmark described here was originally developed for evaluating our OBII system [1]. However our approach could be applied to evaluate other Semantic Web systems in general. In this paper we present the various workload components that are of general interest. We also discuss wherever applicable how they can be further generalized. Specifically we make the following two technical contributions in this paper.

1. We design and implement an algorithm to generate a graph of ontologies defined by parameters like diameter, average out-degree of node ontology, number of paths having a diameter length, number of terminal ontologies, number of maps etc. Thereafter we generate mapping ontology axioms that conform to a subset of OWL DL.

2. We use these in conjunction with an approach to generate synthetic domain ontologies, synthetic data sources and synthetic queries in order to provide a complete Semantic Web workload.

The rest of the paper is organized as follows: In section 2 we provide a background about the related work. In Section 3 we define the various steps of data generation process like generation of domain ontologies, data sources, queries and the graph of ontologies to create mapping axioms. We introduce a mapping language for describing maps. In Section 4, we describe the methodology for carrying out an experiment and the performance metrics that can be used for evaluation. In Section 5, we conclude and discuss future work.

2 Background

The LUBM [2] is an example of a benchmark for Semantic Web knowledge base systems with respect to use in large OWL applications. It makes use of a university domain workload for evaluating systems with different reasoning capabilities and storage mechanisms. Li Ma et. al [3] extend the LUBM so that it can support both OWL Lite and OWL DL (except Tbox with cyclic definition and Abox with inequality definition). However LUBM and extended LUBM use a single domain/ontology namely the university domain comprising students, courses, faculty etc. We need workloads comprising multiple interrelated ontologies.

Tempich and Volz [4] perform statistical analysis of the available Semantic Web ontologies and derive important parameters which could be used to generate synthetic ontologies. T D. Wang et. al [5] have conducted a more recent survey on OWL ontologies and RDFS schemas to perform analysis over the statistical data and report some important trends. The latter are used to determine if there are interesting trends in modeling practices, OWL construct usages and OWL species utilization. These works can be used in determining reasonable parameters for Semantic Web benchmarks but do not present benchmarks in themselves.

There has been some prior work on benchmarking DL systems. Horrocks and Patel-Schneider [6] use a benchmark suite comprising four kinds of tests: concept satisfiability tests, artificial Tbox classification tests, realistic Tbox classification tests and synthetic Abox tests. The TBox refers to the intentional knowledge of the domain (similar to an ontology) and the ABox contains extensional knowledge. Elhaik et. al. [7] provide the foundations for generating random Tboxes and Aboxes. The satisfiability tests compute the coherence of large concept expressions without reference to a Tbox. However, these approaches neither create OWL ontologies nor SPARQL queries and only focus on a single ontology at a time.

Garcia-Castro and Gomez-Perez [8] provide a benchmark suite for primarily evaluating the performance of the methods provided by the WebODE ontology management API. Although their work is very useful in evaluating ontology based tools it provides less information on benchmarking knowledge base systems.

J. Winick and S. Jamin [9], present an Internet topology generator which creates topologies with more accurate degree distributions and minimum vertex covers as compared to Internet topologies. Connectivity is one of the fundamental characteris-

tics of these topologies. On the other hand while considering a Semantic Web of ontologies there could be some ontologies not mapping to any other ontology thereby remaining disconnected from the graph.

3 Data Generation

We now describe the process of generating several types of synthetic workloads to represent a wide variety of situations. While generating the data set the user is given the freedom to modify the independent parameters while the rest essentially serve as controls whose values are dependent on the nature of applications, like information integration etc. The characteristics of a domain ontology and a map ontology are clearly demarcated in that the former does not have any import statements and a map inherits the axioms of the two ontologies being mapped. This approach is equivalent to having a set of ontologies some of which inherit the axioms of the others. But our approach is very useful in creating the graph of ontologies.

3.1 Generation of Domain Ontologies

We implemented a workload generator that allows us to control several characteristics of our dataset. In generating the synthetic domain ontologies we decided to have on the average 20 classes and 20 properties (influenced by the dominance of small ontologies in the current Semantic Web).

Due to restrictions placed on our OBII system our existing implementation only generates domain ontologies comprising subClassOf and subPropertyOf axioms in order to support taxonomic reasoning. Also, following the statistical analysis of the DAML ontology library [4] we maintain more subClassOf axioms than subPropertyOf axioms. We designate these ontologies as simple ontologies. But however we can easily enhance the degree of expressivity to OWL DL or OWL Lite by including complex axioms like unionOf, intersectionOf, inverseOf etc; because the classes/properties used in our ontology are synthetic without possessing any intuitive semantics. Also, there has been some related work like the Artificial Tbox Classification tests of Horrocks and Patel-Schneider [6] for benchmarking DL systems.

To create a domain ontology, we randomly establish subClassOf and subPropertyOf relationships across classes and properties respectively. The class and property taxonomy have an average branching factor of 4 and an average depth of 3.

3.2 Generation of the graph of interlinked ontologies

We consider a directed graph of interlinked ontologies, where every edge is a map from the source ontology to the target ontology. This map ontology comprises a set of mapping axioms. We describe the following terms for discussing such a graph -

- Diameter: The length of the longest path in the graph

- Whether the node is a terminal node i.e. has a zero out-degree. Before the map is created, we determine the number of terminal nodes and randomly mark those many domain ontologies as terminal. The algorithm is so designed, that it prevents a non-terminal node from attaining a zero out-degree. Also, there could be some terminal nodes with a zero in-degree, thereby disconnecting them from the graph.
- Out-path length: The length of the longest outgoing path of a node
- In-path length: The length of the longest incoming path to a node

The inputs to the graph creation algorithm are the number of ontologies, average out-degree of the nodes and diameter of the graph. There is a parameter – *longPaths* which indicates the number of paths having a diameter length. This parameter has been hard coded to 1 because we need to have at least one path of diameter length. The algorithm usually creates additional paths having a diameter length.

Another important parameter is the total number of maps. We show how this can be calculated from other parameters.

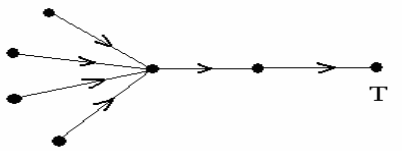
Let

maps – total number of maps
out – average out-degree
onts – total number of ontologies
term – total number of terminal ontologies

Parameters like *maps*, *out* and *term* are interrelated in that *maps* is approximately equal to the product of non terminal ontologies and *out*. Hence we have -

$$(onts - term) * out \cong maps \quad (1)$$

However we do not provide *term* as an input parameter. We show how a reasonable value can be computed from other parameters. We can express *maps* as the product of *term* and *diameter*. The number of maps is at least equal to this product. This is because the in-path length of a terminal node is equal to the diameter. There could be more maps, in situations where more than one diameter length path leads to a terminal node as explained below –



As shown above, the terminal node (marked 'T') has 4 paths of diameter length (diameter is 3) leading to it, effectively yielding more maps. Hence the equation below is desirable but not a requirement. Given that we prefer graphs that will branch out we will use -

$$term \cong maps / diameter \quad (2)$$

Substituting (2) in (1) we get

$$(onts - maps / diameter) * out \cong maps$$

$$onts * out - (maps * out) / diameter \cong maps$$
$$onts * out \cong maps + (maps * out) / diameter$$
$$onts * out * diameter \cong maps * diameter + maps * out$$
$$onts * out * diameter \cong maps * (diameter + out)$$

$$maps \cong (onts * out * diameter) / (diameter + out) \quad \textbf{(3)}$$

Also, by substituting (3) in (2)

$$term \cong (onts * out) / (diameter + out) \quad (\textbf{4})$$

Steps of the Algorithm

1. At the outset determine the number of terminal nodes using the equation- (4) above. Then randomly mark those many domain ontologies as terminal.
2. Thereafter create a path of diameter length. This ensures that there is at least one path of length equal to that of diameter.
3. For every non-terminal ontology, randomly select its out-degree which falls within some range of the specified average out-degree. This range extends by one half of the specified average out-degree on its either side. We choose a uniform distribution for generating a random number. Thereafter randomly select as many target ontologies as the chosen out-degree. The target ontology could be either terminal or non terminal. The sources and the target ontologies will eventually be used for creating mapping axioms.
4. While creating a map ontology between a source and a target certain constraints have to be satisfied which are as follows
    i. The in-path length of the source should be less than the diameter in order to prevent the creation of a path of length greater than the diameter.
    ii. The target should be different from the source
    iii. There shouldn't already exist a direct mapping between the source and the target
    iv. The target should not be among those ontologies from which the source could be visited. This prevents the creation of any cycles in the graph. This is a requirement for OBII, which could be relaxed for other systems.
    v. With the given source and the selected target a transitive path of length greater than the diameter shouldn't be created. This means that the in-path length of the source + the out-path length of the target + 1 should not be greater than the diameter.
    vi. If the target is a non-terminal node and by virtue of creating a map between the source and the target, the latter or any of its non-terminal descendants could become a terminal node then it should be avoided. This happens when the in-path length of the source is one less than the diameter.

vii. There sometimes arises a situation, where none of the existing nodes can satisfy the above constraints. This can happen in cases of large diameters and large out-degrees or when the diameter is equal to the number of ontologies. When such a situation arises a new ontology is created to serve as a target. Such ontologies which are dynamically created are termed as fresh ontologies. So the total number of ontologies at the end may be greater than the number of ontologies with which the algorithm began.

5. Once a map ontology is created the attributes of the source and the target have to updated as follows
   i. The source and the set of ontologies from which it can be reached must be added to the set of ontologies from which the target and its descendants can be reached
   ii. The out-degree of the source has to be updated.
   iii. The source must be made the parent of the target
   iv. The target should be made the child of the source
   v. The out-path length of the source and all its ancestors has to updated if applicable
   vi. The in-path length of the target and all its descendants has to be updated if applicable

3.3 Generation of mapping axioms

Once the source and the target ontologies have been identified mapping axioms need to be established. A specific number of terms (classes and properties) from the source ontology are mapped to terms in the target ontology. Since the domain ontologies are randomly chosen while creating a map ontology we expect the latter to reflect a partial overlap between the two. Hence this value has been hard coded to 20% of the total number of classes and properties in the source ontology.

OBII uses the language OWLII [1] which is a subset of OWL DL. This language has been defined as follows.

Definition **OWLII**

i. Let L_{ac} be a DL language where A is an atomic class, and if C and D are classes and R is a property, then $C \sqcap D$ and $\exists R.C$ are also classes.

ii. Let L_a include all classes in L_{ac}. Also, if C and D are classes then $C \sqcup D$ is also a L_a class.

iii. Let L_c includes all classes in L_{ac}. Also, if C and D are classes then $\forall R.C$ is also an L_c class.

iv. OWLII axioms have the form $C \sqsubseteq D$, $A \equiv B$, $P \sqsubseteq Q$, $P \equiv Q$, $P \equiv Q^{-}$, where C is an L_a class, D is an L_c class, A, B are L_{ac} classes and P, Q are properties.

At present we generate mapping axioms that fall strictly within OWLII. The limited expressivity of OWLII prevents generating inconsistent axioms, but when extended to

more expressive axioms we can incorporate a consistency check to the ontology generation process.

In what follows we first describe how this is implemented in our current system and how it can be easily extended to OWL-DL. We create each mapping axiom by essentially generating an OWL parse tree with the root node being a subclass operator. Then based on a user supplied frequency table of various OWL constructors the tree is expanded by using named classes or owl constructors. The frequency table allows the users to specify a ratio of various owl constructors which they expect to have in their mapping axioms.

Our algorithm recursively builds the parse tree based on the above and terminates by choosing named classes for all the remaining operands when the maximum length of a mapping axiom is reached. If there doesn't exist a mapping between a pair of ontologies, it simply means that the latter are not related and represent different domains. Such a landscape truly reflects the nature of semantic web comprising groups of interrelated ontologies as well as lone ontologies. Thus answering a query demands being selective about particular data sources instead of scanning the entire data set. Our OBII system [1] uses the concept of "rel-files" in order to select only those data sources which contain relevant information.

Note: In the above approach we essentially restrict the axiom generation to remain within OWLII by using certain constructors in either the subject or the object position of an axiom. This is done because our current implementation is geared towards data for OBII system. However, if we lift these restrictions and allow for any constructors to be on either side of the tree, we can generate axioms that are OWL-DL.

3.4 Generation of data sources

A specified number of data sources are generated for every domain ontology. Every data source comprises ABox assertions with named classes/properties. For every source a particular number of classes and properties are used for creating triples. These triples are added to the source ontology being created. The number of classes and properties to be used for creating triples can be controlled by specifying the relevant parameters. With our current configuration the average data source has 75 triples. Considering the sparse landscape of the number of classes/properties from an ontology which are actually instantiated [10] and also due to the lack of knowledge about the prospective manifestation of the actual semantic web we have currently chosen to instantiate 50% of the classes and 50% of the properties of the domain ontology. But however this can be easily modified to suit the nature of application.

3.5 Generation of Queries

Our query generation process generates SPARQL queries from a given set of ontologies. Currently we support single ontology queries i.e. queries that have predicates from a single namespace. This approach can be extended to multi ontology queries quite easily. In our current approach we randomly choose an ontology from a set of

ontologies to be the query ontology. These queries are conjunctive in nature as in the conjunctive query language of Horrocks and Tessaris [11]. We then randomly generate a set of query predicates. The number of predicates for each query is determined by a user specified parameter. We generate the queries based on the following policies:

1. We choose the first predicate from the classes of the query ontology.
2. We bias the next predicate to have a 75% (modifiable) chance of being one of the properties of the query ontology in order to achieve some degree of control over query selectivity.
3. In order to generate interesting queries that require some joins between query predicates, we need to have variables that are shared by at least two predicates of a given query. In order to guarantee this shared variable, when generating a new predicate we can use one variable from the previous predicate that has been generated. If the new predicate is unary we use the variable from the previous predicate and if it is binary in addition to the "used" variable we also create a fresh one. Furthermore in choosing the position of the "used" variable in a new binary predicate that is being created, on the average we choose to put it in the subject position 50% of the time and in the object position 50% of the time. This ensures that the former is equally likely to be in the subject as well as object position of connected triples.
4. If the query we generate is a single predicate query we make all the variables distinguished. For any other queries we make on the average 2/3rd of the variables distinguished and the rest non-distinguished.
5. We bias the introduction of a constant in a query predicate with a chance of 10%.

The above policy reflects our desire to have a simplistic query generation approach that can generate queries that are useful in measuring a system's performance. It allows us to generate queries with a decent mix of classes, properties and individuals.

Note: Every conjunct/constant added to the query makes it more selective. With a diverse data set and randomly generated queries we obtain a wide range in the degree of query selectivity.

4 Experimental Methodology

We present here our methodology of setting up an experiment for OBII and also the performance metrics that could be used for evaluation.

We feel that the most significant parameters that should be investigated are the number of ontologies, data sources, out-degree and diameter. A configuration is denoted as: nO-nD-nS where nO is number of ontologies, nD is diameter and nS is number of sources that commit to an ontology.

Metrics like Load Time, Repository Size, Query Response Time, Query Completeness and Soundness could serve as good candidates for performance evaluation [2].

Load Time: This could be calculated as the time taken to load the Semantic Web space: domain and map ontologies and the selected data sources.
Repository Size: This refers to the resulting size of the repository after loading the benchmark data into the system. Size is only measured for systems with persistent storage and is calculated as the total size of all files that constitute the repository. Instead of specifying the occupied disk space we could express it in terms of the configuration size.
Query Response Time: We recommend this to be based on the process used in database benchmarks where every query is consecutively executed on the repository for 10 times and then the average response time is calculated.
Query Completeness and Soundness: With respect to queries we say a system is complete if it generates all answers which are entailed by the knowledge base. However on the Semantic Web partial answers will also be acceptable and hence we measure the degree of completeness of each query as a percentage of the entailed answers that are returned by the system. On similar lines we measure the degree of soundness of each query as the percentage of the answers returned by the system that are actually entailed. On small data configurations, the reference set for query answers can be calculated by using state of the art DL reasoners like Racer and FaCT. For large configurations we can use partitioning techniques such as those of Guo and Heflin [12].

5 Conclusion and Future Work

Initial inspection has shown that our approach creates reasonable ontology graphs in that they are consistent with our input parameters and also have a good path length distribution from 0 to diameter. The graph topologies for some of the configurations are as follows. The nodes in the graph represent the ontologies and the links represent the mappings. A configuration is denoted by the following triple: "No. of ontologies – Outdegree – Diameter".

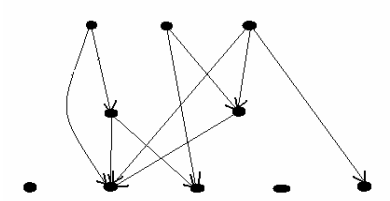


Fig. 1a. 10 -2- 2

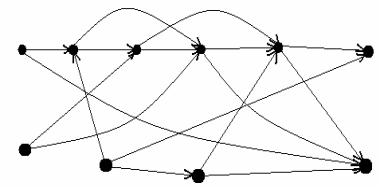


Fig. 1b. 10 – 2 - 5

There are some nodes in Fig. 1a which are disconnected from the graph. These are terminal nodes with zero in-degree. In the actual semantic web there could be such ontologies which do not map to any ontology and remain isolated.

In this paper we have discussed our approach for developing a benchmark for a complete synthetic workload. In any kind of benchmark there is some tradeoff between realism and in being simple and sufficient. Our approach is simple but could be easily generalized to support more expressive domain ontologies.

We have also introduced a new methodology for creating a graph of multiple interrelated ontologies that could be used by distributed query systems like OBII. The graph can be controlled effectively by parameters like diameter and average out-degree of the nodes. We could incorporate additional variables to represent in-degree and out-degree distributions where a few ontologies serve as "hubs" with very high out-degree and in other cases as "authorities" with a very high in-degree.

A single workload is incapable of evaluating different knowledge base systems. But our workload can be easily scaled to various configurations for the purpose of evaluation. This might encourage the development of more scalable reasoners in the near future.

It would be useful to allow the user to specify the distribution of RDFS, OWL Lite, OWL DL ontologies. Furthermore, we intend to conduct an initial experiment for comparing OWL reasoners such as Sesame, KAON2, Minerva and OWLIM.

References

1. A. Qasem, D. A. Dimitrov, J. Heflin. Efficient Selection and Integration of Data Sources for Answering Semantic Web Queries. In *New Forms of Reasoning Workshop, ISWC 2007.*
2. Y. Guo, Z. Pan, and J. Heflin. LUBM: A benchmark for owl knowledge base systems. *Journal of Web Semantics*, 3(2):158–182, 2005.
3. L. Ma, Y. Yang, Z. Qiu, G, Xie and Y. Pan. Towards A Complete OWL Ontology Benchmark. In *Proc. of the third European Semantic Web Conference.(ESWC 2006)*, 2006
4. Tempich, C. and Volz, R. Towards a benchmark for Semantic Web reasoners – an analysis of the DAML library. In *Workshop on Evaluation on Ontology Based Tools, ISWC2003*.
5. T D. Wang, B. Parsia and J. Hendler. A Survey of the Web Ontology Landscape. In *Proc. of the 5th International Semantic Web Conference. (ISWC 2006)*, 2006
6. I. Horrocks and P. Patel-Schneider. DL Systems Comparison. In Proc. Of the 1998 Description Logic Workshop (DL' 98), 1998.
7. Q. Elhaik, M-C Rousset and B. Ycart. Generating Random Benchmarks for Description Logics. In Proc. of the 1998 Description Logic Workshop (DL' 98), 1998.
8. R. Garcia-Castro and A. Gomez-Perez. A Benchmark Suite for Evaluating the Performance of the WebODE Ontology Engineering Platform. In Proc. of the 3rd International Workshop on Evaluation of Ontology-based Tools, 2004.
9. J. Winick and S. Jamin. Inet-3.0: Internet Topology Generator. In University of Michigan Technical Report CSE-TR-456-02.
10. Z. Pan, A. Qasem, J. Heflin. An Investigation into the Feasibility of the Semantic Web. In Proc. of the Twenty First National Conference on Artificial Intelligence (AAAI 2006), Boston, USA, 2006. pp. 1394-1399.
11. I. Horrocks and S. Tessaris. A conjunctive query language for description logic aboxes. In *AAAI/IAAI*, pages 399–404, 2000.
12. Guo Y. and Heflin J. Document-Centric Query Answering for the Semantic Web. 2007 IEEE/WIC/ACM International Conference on Web Intelligence (WI' 07), 2007.
13. B. Grosof, I. Horrocks, R. Volz, and S. Decker. Description logic programs: Combining logic programs with description logic. In *Proceedings of WWW2003*, Budapest, Hungary, May 2003. World Wide Web Consortium.

A Panoramic Approach to Integrated Evaluation of Ontologies in the Semantic Web

Sourish Dasgupta, Deendayal Dinakarpandian, Yugyung Lee
School of Computing and Engineering
University of Missouri-Kansas City,
Missouri, USA
{sdwb7, dinakard, leeyu}@umkc.edu

Abstract. As the sheer volume of new knowledge increases, there is a need to find effective ways to convey and correlate emerging knowledge in machine-readable form. The success of the Semantic Web hinges on the ability to formalize distributed knowledge in terms of a varied set of ontologies. We present Pan-Onto-Eval, a comprehensive approach to evaluating an ontology by considering its structure, semantics, and domain. We provide formal definitions of the individual metrics that constitute Pan-Onto-Eval, and synthesize them into an integrated metric. We illustrate its effectiveness by presenting an example based on multiple ontologies for a University.

Keywords: Ontologies, Semantic Web, Open Evaluation, Ontology Ranking

1 Introduction

An important goal of the Semantic Web [1] is to enable agents to discover knowledge that is distributed across the Web. The distributed knowledge needs to be formalized in the form of ontologies so that relevant subsets may be selected for different purposes. As stated by Sabou et al [2], this necessitates an efficient way to evaluate and rank ontologies. Ontology evaluation is also important for the related problems of ontology discovery, reasoning and modularization [2].

Tartir et al [8] and Sabou et al [2] have compiled various metrics that can be used to evaluate ontologies. Ding et al [3] and Patel et al [4] have proposed evaluation metrics based on a popularity measure that is derived from Google's Page Rank algorithm [5]. A number of semantic search engines like Swoogle [3, 6], OntoKhoj [4] and OntoSelect [7] are based mainly on the popularity measure. Ontology evaluation and ranking can be used for selecting relevant knowledge resources [8] and for determining their quality. Moreover, ontology evaluation can be an efficient basis for comparing several ontologies, as shown in our previous work [9].

Ontology summarization is the extraction of a snapshot of an ontology that contains the most important characteristics of the ontology (concepts and relations that represent the thematic categories of the ontology). Zhang et al [10,11] have introduced ontology summarization for better understanding and improved alignment of similar ontologies. The primary idea underlying their work is the extraction of relevant vocabularies from ontologies based on notions such as RDF[1] sentences and RDF graphs. They have not applied it to the evaluation of ontologies. To our

[1] http://www.w3.org/RDF/

knowledge there has been limited work on the use of ontology summaries for the purpose of ontology evaluation. Another important aspect is with regard to scalability. Current evaluation methodologies are not scalable for a large ontology. An intuitive way to handle this problem might be to modularize ontologies according to usage patterns (Sabou et al [2] and Noy [12]). However, on-the-fly modularization of ontologies based on queries is challenging due to the significant computation cost required for ontology modularization *per se*. This motivated us to use summaries of ontologies as the basis of our evaluation computation instead of dealing with the entire ontology.

In this paper, we propose a novel way of evaluating ontologies based on our ontology summarization technique [13] that focuses on multiple semantic dimensions of ontologies. In view of the extensive diversity of ontologies, we need an integrated approach to ontology evaluation that considers its domain as well as structural and semantic perspectives.

2. Related Work

Several research efforts have tried to classify different methods for evaluating ontologies based on these objectives [14,15]. Some work (Swoogle [3,6], OntoSelect [7] and OntoKhoj [4]) focus on measurement of the authoritativeness of an ontology by utilizing relevant and important cross-references of the ontology and rank them similar to PageRank [5]. However, Alani et al [16] pointed out that cross-references between ontologies might not be always available and hence evaluation based solely on this criterion might fail. Furthermore, even though an ontology might be well connected with several other ontologies, they might cover topics differently and have different semantic implications. Thus, the importance of an ontology cannot be captured simply by calculating its degree of reference.

Structural richness is a measure of the topological aspect (depth and height) of an ontology. Tartir et al [8] have termed it as "inheritance richness." This criterion measures how the information is distributed over the entire ontology and determines whether the ontology is domain-specific (the depth is greater than the width) or generic (the width is greater than the depth). Another approach is to determine the significance of a particular concept based on the number of super and sub concepts [16-18]. In [16], two very important metrics have been considered: density measure and centrality measure [18]. Density is determined based on the number of super and sub concepts of the given concept. Centrality is a measure of how far a concept is from the root concept in its hierarchy, relative to the length of the longest path from the root to a leaf node containing the concept. It is assumed that concepts in the center of an ontology are the most representative. This kind of evaluation relies largely on the structural aspect of concepts in ontologies.

Relational richness is a measure that captures how a concept is related to other concepts. According to Tartir et al [8], relational richness of an ontology is defined as the ratio of the number of non-IS-A relations to the total number of relations in the ontology. This definition, however, is somewhat simplistic. It is because this approach does not take into account the roles of concept, domain (subject) or range (object), for

a given relation. A similar concern for relational richness can be found in Sabou et al [2] where no model has been defined. It takes all relations into account regardless of the fact that there may be more than one concept hierarchy in a single ontology. Thus, it is important that the set of relations pertaining to a hierarchy should be treated separately from those in different hierarchies. Otherwise the thematic differences between these hierarchies cannot be correctly captured; this measure cannot properly reflect the perspective of an ontology. Existing studies are limited in measuring the semantics of relations in an ontology. In our model, we take the roles of the concepts involved in relations into consideration and additional categories of relations for ontology evaluation.

3. Proposed Model – Fundamental Concepts

We now present our ontology evaluation model, called Pan-Onto-Eval that builds on our previous work on ontology summarization [13]. Ontology summarization aims to extract a snapshot of an ontology that contains the most important characteristics of the ontology (concepts and relations that represent the thematic categories of the ontology). Our measurement represents a comprehensive perspective on the following four important issues: a) *Triple Centricity*, b) *Theme Centricity*, c) *Structure Centricity* and d) *Domain Centricity*. We hypothesize that all these features are highly related to each other so that an integrated model can serve efficiently as the basis of evaluation metrics. We elaborate on these fundamental concepts below.

a) *Triple Centricity*: This is the central feature of our model. In an ontology O, the relations (denoted by R) can be either IS-A relations (denoted by R^S) exclusively or non-IS-A relations (denoted by R^N): $R^S \subset R$, $R^N \subset R$ and $R^N \cap R^S = \varphi$. Given any non-IS-A relation, a concept can be either a domain concept (DC) or a range concept (RC) depending upon its role in the relation. A concept associated with a non-IS-A relation can be either a *DC* or a *RC*.

Regarding the triple centric evaluation, we say that an ontology is meaningful when there are many diverse relationships, i.e., domain concepts associated with other concepts through diverse relations. Hence we analyze their roles with relations (i.e. whether they are domain or range concepts) and their importance (the measurement of concept importance) described in our work on ontology summarization [13]. Furthermore, we analyze how the range concepts are associated within these domains as the range concepts play an important role, i.e., the information source, to the domain concepts. In this way, we evaluate an ontology from a triple centric perspective that is distinct from other works [8, 16-18].

b) *Theme Centricity:* This refers to the classification of non-IS-A relations in an ontology. This is a measure that efficiently reflects the importance of non-ISA relations in the evaluation of any ontology in terms of relational richness. Tartir et al [8] stated "An ontology that contains many relations other than class-subclass relations is richer than a taxonomy with only class-subclass relationships". Sabou et al [2] considered relations as a primary criterion for the summary extraction of ontologies. However, they concentrated on a quantitative aspect such as the

percentage of non-IS-A vs. IS-A relations [8] and did not take into account how these non-IS-A relations are distributed over an ontology.

In our work, seven broad thematic categories for classification of non-IS-A relations inspired by UMLS [19] have been defined as follows: Compositional, Attributive, Spatial, Functional, Temporal, Comparative and Conceptual. It is evident from the justification provided for the triple centric approach that the relations between domain and range concepts carry different semantic 'senses'. This classification thus provides for better understanding of the thematic categories of the ontology so that it may facilitate effective ontology evaluation and querying. This is because it allows one to map relations existing in query triples to those contained in the ontology.

c) *Structure Centricity*: This measure describes the topology (i.e., shape and size) of concept hierarchies of an ontology. Consider two topologies [8, 9]: The top-shaped hierarchy has a characteristic such that the breadth of class hierarchies decreases as the depth increases. This ontology is more generalized in its thematic category. On the other hand, the pyramidal hierarchy has a characteristic such that the breadth of class hierarchies increases as the depth increases. They are more domain-specific. However, in reality, ontologies have more irregular shape in terms of the breadth-depth ratio. Previous works [8, 9]only consider the average number of sub-classes of a given hierarchy. Thus, this measure would not be appropriate for evaluating diverse structural aspects of ontology. From a structural perspective, we may want to analyze the distribution of non-IS-A relations. If a relation appears at a high level, it might be too abstract. Otherwise, it might be too specific.

d) *Domain Centricity*: An ontology may consist of more than one IS-A hierarchy. Each of these hierarchies might suggest that their *thematic category* (or semantic implication) is different. In other words, each hierarchy contributes differently to the semantics of the ontology as a whole. Each hierarchy consists of some domain concepts typed under their own root; the specific perspective of these hierarchies may be characterized by their relations and range concepts. That is why we analyze the semantic richness of a hierarchy based on the comprehensiveness criterion (in Section 4) and incorporate the measure into an ontology evaluation score. We assume that this approach is more appropriate than taking the ontology as a whole because it considers the semantics and distribution of information across the ontology.

4. Pan-Onto-Eval Metrics

We now formalize our ontology evaluation metrics of the Pan-Onto-Eval. The evaluation metric is defined by considering the following five qualitative aspects of ontology: (1) *Information content*, (2) *Relational Richness* (3) *Inheritance Richness*, (4) *Dimensional Richness*, and (5) *Domain Importance*. In the Pan-Onto-Eval, for a given ontology, we independently analyze each hierarchy that exists under the root of the ontology independently and combine information from multiple hierarchies into information representing the ontology as a whole.

We define the parameters that will be used in the formula:

M: Number of range concepts in H

Mi: Number of selected range concepts with the thematic category i in the summary

N: Number of domain concepts in H
Ni: Number of selected domain concepts in the thematic category i in the summery
Q: Number of the thematic categories of relations in H
Q': Total number of thematic categories (in our model it is seven)
R: Number of non-IS-A relations in H
R_t(RC): number of relations classified under the thematic category t for a range concept RC
R(i): Number of relations selected in the thematic category i in the summary.
R(DC): Number of relations associated with the domain concept DC
S(DC_i) Number of direct sub-concept (children) under the domain concept DC_i in H
α: Normalization function (a sigmoid function is used in the analysis)
K: number of hierarchies in the ontology

1) *Information Content (IC)* measures how well information involving relations *R* is distributed over an IS-A hierarchy *H* in an Ontology *O*. Our hypothesis with regard to *IC* is that a well spread distribution of important relations with respect to domain concepts *DC* in *H* indicates richness of information. For this purpose, we borrow the basic formula for information entropy[20] to determine degree of information content of ontologies. We measure the number of relations in terms of the number of range concepts RC that are associated with the hierarchy *H*.

Information Addition (IA) measures how important a Range Concept (RC) is as compared to other *RCs* associated with a hierarchy. This can be represented as the ratio of the number of observed relations associated with a thematically categorized *RC* to the maximum number of possible relations of the *RC*. The maximum number of possible relations of a *RC* is defined using the pigeon hole principle[2]as follows:

$$IA(RC) = \frac{\sum_{t=1}^{Q} R_t(RC)}{R - M + 1} \quad \textbf{(1)}$$

Entropy of the Hierarchy E(H) is the amount of uncertainty associated with the relational association of the *RC* to the hierarchy *H*. In other words, the overall uncertainty of associated *RCs* can be measured as below.

$$E(H) = -\sum_{i=1}^{M} IA(RC_i) \bullet \log_2 IA(RC_i) \quad \textbf{(2)}$$

We now formally define *Information Content* (*IC)* of an IS-A hierarchy *H* as:

$$IC(H) = R \bullet \alpha \bullet \frac{1}{E(H)} \quad \textbf{(3)}$$

A high value for *IC* implies that the information content of the hierarchy *H* in an ontology is rich due to rich relationships defined in *H*.

2) *Relational Richness (RR)*: This metric measures the degree of important relations in a particular hierarchy of an ontology. We define *RR* for the hierarchy *H* as follows:

[2] http://zimmer.csufresno.edu/~larryc/proofs/proofs.pigeonhole.html

$$RR(H) = \frac{1}{Q} \bullet \sum_{t=1}^{Q} R(t) \quad (4)$$

This metric equation captures the important relations associated with the range concepts that are scanned while generating the summary.

3) *Inheritance Richness (IR)* captures whether the hierarchical (IS-A) relations are rich both structurally as well as in their information content. This is important because a concept may have a rich set of sub-concepts but without carrying much information *per se*. Such cases have been ignored in the metric definition of previous works [8]. We define *IR* of a particular hierarchy *H* as:

$$IR(H) = \frac{1}{N} \sum_{i=1}^{N} S(DG) \bullet R(DG) \quad (5)$$

4) *Dimensional Richness (DR)* measures the richness of the thematic categories of relations in a hierarchy of an ontology. This shows the different ways that an ontology hierarchy can satisfy queries based on their summary content. We formally define *DR* of an IS-A hierarchy *H* as:

$$DR(H) = \frac{Q}{Q'} \sum_{i=1}^{Q} N_i \bullet M_i \quad (6)$$

The first factor of Equation 6 indicates the relative coverage of thematic categories for an ontology. The second factor indicates the richness of all of these categories in terms of the number of important (selected) range concepts and their domain concepts. If the value of *DR* is high then it suggests that the corresponding ontology carries a rich semantic dimensionality with a very high ratio of the identified categories versus the total number of defined categories. It also indicates either a very high density of selected range concepts and/or a very high density of corresponding domain concepts in the ontology summary. This means that the ontology is rich in certain thematic categories and queries based on those categories can be best served.

5) Domain Importance (DMI): This metric provides an insight to the richness of the core domain(s) of interest that a particular hierarchy H_k contains when compared to other hierarchies of the same ontology. This metric is basically a compound metric of the previous three metrics. We define *Domain Factor* (*DMF*) and *Domain Importance* (*DMI*) as follows:

$$DMF(H_k) = IC(H_k) + IR(H_k) + DR(H_k) + RR(H_k) \quad (7)$$

$$DMI(H_k) = \frac{DMF(H_k)}{MAX_{i=1}^{k}(DMF(H_i))} \quad (8)$$

If *DMI* is closer to the maximum possible value, this means that the domain represented by this hierarchy is important compared to other hierarchies.

Ontology Evaluation Score (ρ): For a given ontology O, we analyze the richness of each hierarchy within O separately and according to respective criteria. We can now combine them together into a single model that can effectively evaluate ontologies. In order to combine the individual analysis of hierarchies, we compute it as the product of the average of *DMI* and the maximum *DMF* (the best one). We formalize the ontology evaluation score (denoted by ρ) as follows:

$$\rho(o) = MAX_{i=1}^{k}(DMF(H_i)) \bullet \frac{1}{K} \bullet \sum_{i=1}^{K} DMI(H_i) \quad \textbf{(9)}$$

5. Experimental Results

We analyze three related university ontologies (O_1[3] O_2[4], O_3[5]) and evaluate them according to the proposed model. As preprocessing, we convert the DAML files to OWL using a converting tool[6] and generate summaries. The application is implemented using the Protégé OWL 3.3 beta API on a Windows machine. Table 1 shows the analysis of ontology University-I. We analyze the 9 hierarchies among 11 (denoted as H_i) in the ontology excluding two hierarchies (they have single concept with no relation). Hierarchy H_6 has the highest number of associated non-IS-A relations (12) and the highest number of range concepts (9) while H_5 has the maximum number of domain concepts (5) and the maximum levels.

It is interesting to note that although H_6 and H_7 are structurally and relationally rich than the others yet they have a low *Information Content* (IC). This is because the relations are not distributed evenly throughout the hierarchy and most of the domain-concepts in the hierarchy are weakly associated with range concepts in terms of information distribution. Hierarchy H_5 has the highest *Domain Importance* (DMI) value and thus is considered the best hierarchy of this ontology. This accounts for the high *Inheritance Richness* (IR) score and *Dimensional Richness* (DR) score as compared to other hierarchies and hence shows how important it is to have high-weight relations associated with the concepts (and sub-concepts) of a hierarchy. The contributing factor is the dimensional variety of the summary which reflects the rich categorical coverage of the hierarchy as a whole. This hierarchy is rooted at the domain-concept '*Document*' and covers the *attributive*, *functional* and *temporal* aspects evenly. The next best hierarchy is H_7 rooted at the concept '*Organization*' with the majority of relations falling under the categories *conceptual* and *attributive*. Close to this hierarchy is H_6 rooted at '*Organism*'. The rest of the hierarchies have pretty low *DMI* values. The evaluation score of the University-I (ρ) is 6.109.

Analyzing Table 2 indicates that the University–II ontology is an instantiation of the University-I. It is interesting to see that the new hierarchy (having a single concept

[3] http://www.ksl.stanford.edu/projects/DAML/ksl-daml-desc.daml
[4] http://www.ksl.stanford.edu/projects/DAML/ksl-daml-instances.daml
[5] http://www.cs.umd.edu/projects/plus/DAML/onts/univ1.0.daml
[6] http://www.mindswap.org/2002/owl.shtml

'Chimaera-Export-Enable') adds no richness to the ontology. An important observation is that the best hierarchy in this ontology is H_6 as compared to its parent ontology where the best hierarchy is H_5. This is because of the partial use of the University-I ontology. This leads to a lowering of the *DR* value and the *RR* value of H_5. The evaluation score of the ontology (ρ) is 3.909.

Table 1. Evaluation of University – I

	H_1	H_2	H_3	H_4	H_5	H_6	H_7	H_8	H_9
Number of relations (R)	2	1	3	3	4	12	11	1	3
Number of range concepts (M)	2	1	3	3	4	9	7	1	3
Number of Domain concepts (N)	1	1	1	1	5	4	2	1	1
Information content (IC)	2	1	3	3	4	3	3.52	1	3
Inheritance richness (IR)	0	0	0	0	4	3	1	0	0
Dimensional richness (DR)	0.57	0.14	1.28	1.28	1.7	1.4	3.4	0.14	0.57
Relational richness (RR)	1	1	1	1	1.33	2.4	2.75	1	1.5
Domain factor (DMF)	2.57	2.14	3.28	3.28	8.03	7.05	7.15	2.14	3.07
Domain importance (DMI)	0.29	0.27	0.38	0.38	1	0.87	0.89	0.27	0.37

Table 2. Evaluation of University - II

	H_1	H_2	H_3	H_4	H_5	H_6	H_7	H_8
Number of relations (R)	0	1	3	0	2	6	5	2
Number of range concepts (M)	0	1	3	0	2	6	3	2
Number of Domain concepts (N)	1	1	1	1	5	4	2	1
Information content (IC)	0	1	3	0	2	6	2.9	2
Inheritance richness (IR)	0	0	0	0	0	0	0	0
Dimensional richness (DR)	0	0.14	1.28	0	0.57	1.71	2.85	0.57
Relational richness (RR)	0	1	1	0	1	2	1.25	1
Domain factor (DMF)	0	2.14	3.28	0	2.57	4.71	4.68	2.57
Domain importance (DMI)	0	0.454	0.696	0	0.546	1	0.99	0.546

Table 3. Evaluation of University - III

	H_1	H_2	H_3	H_4	H_5	H_6	H_7
Number of relations (R)	1	3	1	6	2	0	0
Number of range concepts (M)	1	2	1	4	2	0	0
Number of Domain concepts (N)	1	16	2	4	7	2	3
Information content (IC)	1	1.95	1	3.3	2	0	0
Inheritance richness (IR)	0	7	0	8	0	0	0
Dimensional richness (DR)	0.14	0.57	0.14	1.28	0.57	0	0
Relational richness (RR)	1	1	1	1	1	0	0
Domain factor (DMF)	2.14	9.22	2.14	10.83	2.57	0	0
Domain importance (DMI)	0.198	0.851	0.198	1	0.237	0	0

The third ontology, University-III, has been analyzed in Table 3. This ontology is different semantically from the previous two ontologies although there are common concepts among them. This is because the associated relations (and hence the semantic categories) are quite different. H_4 is rooted at '*Person*' and has 4 *DCs*, 4 *RCs*

and 6 Relations. Incidentally, this hierarchy is structurally the best among the seven hierarchies of the ontology. If we compare H_4 with H_2 (rooted at '*Employee*') we will see the number of *RCs* and relations in H_2 are smaller compared to H_4. Although the number of *DCs* in H_2 is 16 (four times that of H_4) yet the *IR* value (7) is lower than that of H_4 (8). This is because most of the inheritances in H_2 are void relationally (3 Relations and 2 *RCs*). This means they have no semantic importance although they are very rich structurally. The second best structurally rich hierarchy is H_5 (7 *DCs*). But this hierarchy has low *DMI* due to low dimensional richness, in spite of *IC* being high. The other important factor for such a low *DMI* is that the relations are associated with the leaf concepts of the hierarchy and hence the *IR* value is 0 (compared to 8 of H_4 and 7 of H_5). The evaluation score of the University-III (ρ) is 4.567.

We give a comparative analysis of these three ontologies in Figure 1 showing the break-up of the average contribution of each of the metrics for the final evaluation score.

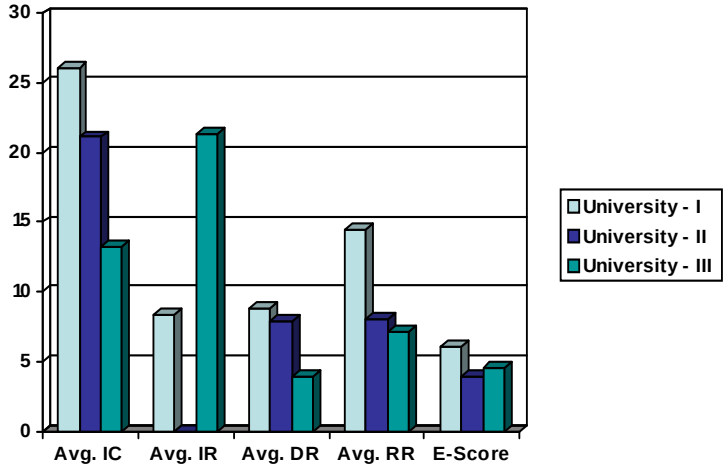


Fig. 1. Comparison of the three ontologies (IC, IR, DR, RR are scaled by factor 10)

7. Conclusion

This paper has presented Pan-Onto-Eval, a comprehensive approach to evaluating an ontology by considering various aspects like structure, semantics, and domain. The main contribution of this paper is a formal treatment of the model for an automated and integrated evaluation of ontologies. The experimental results of the university ontologies demonstrate the essence and benefits of the proposed model. This work is limited by a lack of rigorous evaluation by experts. The summarization technique that is an important basis could have been fully explored and the thematic categories may further be expanded for real world applications. Overall, the model has great potential on evaluation and selection of distributed knowledge in the Semantic Web.

References

1. Berners-Lee, T., J. Hendler, and O. Lassila, *The Semantic Web.* Scientific American, 2001. **284**(5): p. 34-43.
2. M. Sabou, et al. *Ontology Selection; Ontology Evaluation on the Real Semantic Web*. in *the Evaluation of Ontologies for the Web (EON)*. 2006.
3. L. Ding, et al. *Finding and Ranking Knowledge on the Semantic Web* in *the 4th International Semantic Web Conference*. 2005.
4. C. Patel, et al. *OntoKhoj: A Semantic Web Portal for Ontology Searching, Ranking and Classification*. in *the 5th International ACM Workshop on Web Information and Data Management*. 2003.
5. L. Page, et al., *The PageRank Citation Ranking: Bringing Order to Web* 1998, Stanford.
6. L. Ding, et al. *Swoogle: A Search and Metadata Engine for the Semantic Web* in *13th ACM International Conference on Information and Knowledge Management*. 2004.
7. P. Buitelaar, T. Eigner, and T. Declerck. *Ontoselect: A Dynamic Ontology Library with Support for Ontology Selection*. in *the International Semantic Web Conference*. 2004. Hiroshima, Japan.
8. S. Tartir, et al. *OntoQA: Metric-Based Ontology Quality Analysis*. in *IEEE Workshop onKnowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*. 2005.
9. K. Supekar, C. Patel, and Y. Lee. *Characterizing Quality of Knowledge on Semantic web*. in *the AAAI Florida AI Research Symposium* 2004.
10. X. Zhang, G. Cheng, and Y. Qu. *Ontology Summarization Based on RDF Sentence Graph*. in *16th International World Wide Web Conference*. 2007.
11. X. Zhang, H. Li, and Y. Qu. *Finding Important Vocabulary within Ontology*. in *1st Asian Semantic Web Conference (ASWC)*. 2006.
12. Noy, N.F., *Evaluation by Ontology Consumers* IEEE Intelligent Systems, 2004. **19**(4): p. 74-81.
13. S. Dasgupta and Y. Lee, *Relation Oriented Ontology Summerization*. 2007, University of Missouri - KC.
14. J. Brank, M. Grobelnik, and D. Mladenic. *A Survey of Ontology Evaluation Techniques*. in *Conference on Data Mining and Data Warehouses, SiKDD*. 2005. Ljubljana, Slovenia.
15. J. Hartmann, et al. *Methods for ontology evaluation*. in *Knowledge Web Deliverable D1.2.3*. 2005.
16. H. Alani, C. Brewster, and N. Shadbolt:. *Ranking Ontologies with AKtiveRank*. in *5th International Semantic Web Conference*. 2006.
17. H. Alani and C. Brewster. *Metrics for Ranking Ontologies*. in *15th International Conference for World Wide Web*. 2006. Edinburgh, UK.
18. H. Alani and C. Brewster. *Ontology Ranking based on the Analysis of Concept Structures*. *International Conference on Knowledge Capture* 2005.
19. V. Lopez, M. Pasin, and E. Motta. *AquaLog: An Ontology Portable Question Answering System for the Semantic Web*. in *European Semantic Web Conference (ESWC)*. 2005.
20. Shannon, C.E., *A Mathematical Theory of Communication.* Bell System Technical Journal, 1948. **27**: p. 379-423, 623-656.

Sample Evaluation of Ontology-Matching Systems

Willem Robert van Hage[1,2], Antoine Isaac[1], and Zharko Aleksovski[3]

[1] Vrije Universiteit, Amsterdam
[2] TNO Science & Industry, Delft
[3] Philips Research, Eindhoven
{wrvhage,aisaac,zharko}@few.vu.nl

Abstract. Ontology matching exists to solve practical problems. Hence, methodologies to find and evaluate solutions for ontology matching should be centered on practical problems. In this paper we propose two statistically-founded evaluation techniques to assess ontology-matching performance that are based on the application of the alignment. Both are based on sampling. One examines the behavior of an alignment in use, the other examines the alignment itself. We show the assumptions underlying these techniques and describe their limitations.

1 Introduction

The advent of the Semantic Web has led to the development of an overwhelming number[4] of ontologies. Therefore, cross-referencing between these ontologies by means of ontology matching is now necessary. Ontology matching has thus been acknowledged as one of the most urgent problems for the community, and also as one of the most scientifically challenging tasks in semantic-web research.

Consequently, many matching tools have been proposed, which is a mixed blessing: *comparative* evaluation of these tools is now required to guide both ontology-matching research and application developers in search of a solution. One such effort, the Ontology Alignment Evaluation Initiative[5] (OAEI) provides a collaborative comparison of state-of-the-art mapping systems which has greatly accelerated the development of high-quality techniques. The focus of the OAEI has been mainly on comparing mapping techniques for research.

Good evaluation of ontology-matching systems takes into account the purpose of the alignment.[6] Every application has different requirements for a matching system. Some applications use rich ontologies, others use simple taxonomies. Some require equivalence correspondences, others subsumption or even very specific correspondences such as artist-style or gene-enzyme. Also, the scope of concepts and relations is often determined by unwritten application-specific rules (*cf.* [2]). For example, consider the subclass correspondence between the concepts Gold and Jewelry. This correspondence holds if the scope of Gold is limited to the domain of jewelry. Otherwise the two would just be related terms. In either case, application determines relevance.

[4] `http://swoogle.umbc.edu` indexes over 10,000 ontologies by 2007.

[5] `http://oaei.ontologymatching.org`

[6] In this paper we use the definitions as presented in [1]: An ontology matching system produces a set of correspondences called an alignment.

The best way to evaluate the quality of an alignment is trough extensive practical use in real-world applications. This, however, is usually not feasible. The main reason for this is usually lack of time (*i.e.* money). Benchmarks and experiments using synthesized ontologies can reveal the strengths and weaknesses of ontology-matching techniques, but disregard application-specific requirements. Therefore, the second best option is to perform an evaluation that mimics actual usage. Either by performing a number of typical usage scenarios or by specifying the requirements an application has for the alignment and then testing whether these requirements are met. The final measure for system performance in practice is user satisfaction. For the evaluation of matching systems, this means that a set of correspondences is good if users are satisfied with the effect the correspondences have in an application.

Most current matching evaluation metrics simulate user satisfaction by looking at a set of assessed correspondences. For example, Recall expresses how many of the assessed correspondences are found by a system. This has two major problems. (i) Some correspondences have a larger logical consequence than others. That is to say, some correspondences subsume many other correspondences, while some only subsume themselves. This problem is addressed quite extensively in [3] and [4]. (ii) Correct correspondences do not automatically imply happy users. The impact of a correspondence on system performance is determined not only by its logical consequence, but also by its relevance to the user's information need. A correspondence can be correct and have many logical implications, but be irrelevant to the reasoning that is required to satisfy the user. Also, some correspondences have more impact than others.

In the following sections we propose two alternative approaches to include relevance into matching evaluation, one based on *end-to-end evaluation* (Sec. 2) and one based on *alignment sample evaluation* (Sec. 3). Both approaches use sample evaluation, but both what is sampled and the sample selection criteria are different. The former method uses sample queries, disregarding the alignment itself, and hence providing objectivity. The latter uses sample sets of correspondences which are selected in such a way that they represent different requirements of the alignment. We investigate the limitations of these statistical techniques and the assumptions underlying them. Furthermore, we calculate upper bounds to the errors caused by the sampling. Finally, in Sec. 4 we will demonstrate the workings of the latter of the two evaluation methods in the context of the OAEI 2006 food track.

2 End-to-end Evaluation

This approach is completely system-performance driven, based on a sample set of representative information needs. The performance is determined for each trial information need, using a measure for user satisfaction. For example, such an information need could be *"I would like to read a good book about the history of steam engines."* and one could use F-score or the Mean-Reciprocal Rank[7] of the best book in the result list, or the time users spent to find an answer. The set of trials is selected such that it fairly represents different kinds of usage, *i.e.* more common cases receive more trials. Real-life topics should get adequate representation in the set of trials. In practice the trials

[7] One over the rank of the best possible result, *e.g.* 1/4 if the best result is the fourth in the list.

are best constructed from existing usage data, such as log files of a baseline system. Another option is to construct the trials in cooperation with domain experts. A concrete example of an end-to-end evaluation is described in [5]. In their paper, Voorhees and Tice explicitly describe the topic construction method and the measure of satisfaction they used for the end-to-end evaluation of the TREC-9 question-answering track. The size and construction methods of test sets for end-to-end retrieval have been investigated extensively in the context of information retrieval evaluation initiatives such as TREC [6], CLEF, and INEX[8]. When all typical kinds of usage are fairly represented in the sample set, the total system performance can be acquired by averaging the scores.[9] To evaluate the effect of an ontology alignment, one usually compares it to a baseline alignment in the context of the same information system. By changing the alignment while keeping all other factors the same, the only thing that influences the results is the alignment. The baseline alignment can be any alignment, but a sensible choice is a trivial alignment based only on simple lexical matching.

Comparative End-to-end Evaluation

n	number of test trials (*e.g.* information system queries) in the evaluation sample
A, B	two ontology-matching systems
A_i	outcome of the evaluation metric (*e.g.* Semantic precision [3]) for the i-th test trial for system A
$I[A_i > B_i] = \begin{cases} 1 & A_i > B_i \\ 0 & A_i \leq B_i \end{cases}$	interpretation function that tests outperformance
$S_+ = \sum I[A_i > B_i]$	number of trials for which system A outperforms system B

To compare end-to-end system performances we determine whether one system performs better over a significant number of trials. There are many tests for statistical significance that use pairwise comparisons. Each test can be used under different assumptions. A common assumption is the normal distribution of performance differences: small differences between the performance of two systems are more likely than large differences, and positive differences are equally likely as negative differences. However, this is not very probable in the context of comparative evaluation of matching systems. The performance differences between techniques are usually of a much greater magnitude than estimation errors. There are many techniques that improve performance on some queries while not hurting performance on other queries. This causes a skewed distribution of the performance differences. Therefore, the most reliable test is the Sign-test [8, 9]. This significance test only assumes that two systems with an equal performance are equally likely to outperform each other for any trial. It does not take

[8] respectively `http://trec.nist.gov`, `http://www.clef-campaign.org`, and `http://inex.is.informatik.uni-duisburg.de`

[9] A more reliable method for weighted combination of the scores that uses the variance of each performance measurement is described in [7].

into account how much better a system is, only in how many cases a system is better. The test gives reliable results for at least 25 trials. It needs relatively large differences to proclaim statistical significance, compared to other statistical tests. This means statistical significance calculated in this way is *very* strong evidence.

To perform the Sign-test on the results of systems A and B on a set of n trials, we compare their scores for each trial, $A_1, \ldots, A_n$ and $B_1, \ldots, B_n$. Based on these outcomes we compute S_+, the total the number of times A has a better score than B. For example, the number of search queries for which A retrieves better documents than B. The null-hypothesis is that the performance of A is equal to that of B. This hypothesis can be rejected at a confidence level of 95%[†] if

$$\frac{2 \cdot S_+ - n}{\sqrt{n}} > 1.96$$

For example, in the case of 36 trials, system A performs significantly better than system B when it outperforms system B in at least 23 of the 36 trials.

3 Alignment Sample Evaluation

Another evaluation approach is to assess the alignment itself. However, in practice, it is often too costly to manually assess all the correspondences. A solution to this problem is to take a small *sample* from the whole set of correspondences [10]. This set is manually assessed and the results are generalized to estimate system performance on the whole set of correspondences. As opposed to the elegant abstract way of evaluating system behavior provided by *end-to-end evaluation*, *alignment sample evaluation* has many hidden pitfalls. In this section we will only investigate the caveats that are inherent to sample evaluation. We will not consider errors based on non-sampling factors such as judgement biases, peculiarities of the ontology-matching systems or ontologies, and other unforeseen sources of evaluation bias.

Simple Random Sampling

p	true proportion of the samples produced that is correct (unknown)
n	number of sample correspondences used to approximate p
$\hat{P}$	approximation of p based on a sample of size n
δ	margin of error of $\hat{P}$ with 95% confidence

The most common way to deal with this problem is to take a small *simple random sample* from the whole set of correspondences. Assessing a set of correspondences can be seen as classifying the correspondences as *Correct* or *Incorrect*. We can see the output of a matching system as a *Bernoulli random variable* if we assign 1 to every *Correct* correspondence and 0 to each *Incorrect* correspondence it produces. The true

[†] About 95% of the cases fall within 1.96 times the standard deviation from the mean of the normal or binomial distribution. In the derivations we use 2 instead of 1.96 for the sake of simplicity. This guarantees a confidence level of more than 95%.

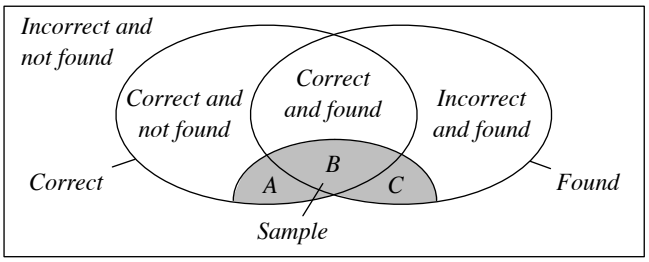


Fig. 1. Venn diagram to illustrate sample evaluation. $A \cup B$ is a sample of the population of Correct correspondences. $B \cup C$ is a sample of the population of Found correspondences.

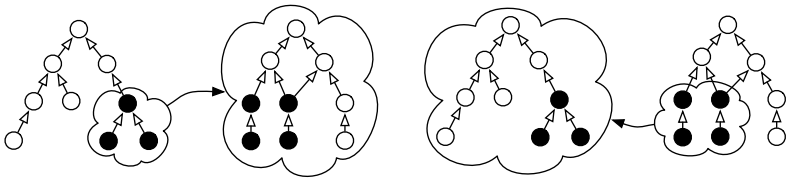


Fig. 2. Concepts to consider when creating a sample for Recall evaluation based on a topic. Black concepts are "on topic", white concepts "off topic". For example, the black concepts have something to do with steam engines and the white concepts do not. Concepts to consider for sample correspondences are marked by clouds. This avoids bias against cross-topic correspondences.

Precision of a system is the probability with which this random variable produces a 1, p. We can approximate this p by the proportion of 1's in a *simple random sample* of size n. With a confidence of 95% this approximation, $\hat{P}$, lies in the interval:

$$\hat{P} \in [p - \delta, p + \delta] \quad \text{where} \quad \delta = \frac{1}{\sqrt{n}} \tag{1}$$

Both Precision and Recall can be estimated using samples. In the case of Precision we take a random sample from the output of the matching system, *Found* in Fig. 1. In this figure the sample for Precision is illustrated as $B \cup C$. The results for this sample can be generalized to results for the set of all *Found* correspondences. In the case of Recall we take a random sample from the set of all correct correspondences, *Correct* in Fig. 1. The sample for Recall is illustrated as $A \cup B$. The results for this sample can be generalized to results for the set of all *Correct* correspondences.

A problem with taking a random sample from all *Correct* correspondences is it is unknown which correspondences are correct and which are incorrect a priori. A proper random sample can be taken by randomly selecting correspondences between all possible correspondences between concepts from the two aligned ontologies, *i.e.* a subset of the cartesian product of the sets of concepts from both ontologies. Each correspondence has to be judged to filter out all incorrect correspondences. This can be very time-consuming if there are relatively few valid correspondences in the cartesian product. The construction time of the sample of correct correspondences can be reduced

by only judging parts of the ontologies that have a high topical overlap. For example, one can only consider all correct mappings between concepts having to do with steam engines. (*cf. e.g.* [11]) It is important to always match concepts about a certain topic in ontology X to *all* concepts in ontology Y, and all concepts about the same topic in ontology Y to *all* concepts in ontology X. This is illustrated in Fig. 2. This avoids a bias against correspondences to concepts outside the sample topic.

There are two caveats when applying this approximation method. (i) A sample of correct mappings constructed in this way is arbitrary, but not completely random. Correspondences in the semantic vicinity of other correspondences have a higher probability of being selected than "loners". This means ontology matching techniques that employ structural aspects of the ontologies are slightly advantaged in the evaluation. (ii) The method works under the assumption that correspondences inside a topic are equally hard to derive as correspondences across topics.

Stratified Random Sampling

N	size of the entire population, *e.g.* the set of all correct correspondences
h	one stratum of the entire population
N_h	size of stratum h
n_h	number of sample correspondences used to approximate p of stratum h
$\hat{P}_h$	approximation of p for the correspondences in stratum h

A better way than *simple random sampling* to perform sample evaluation is *stratified random sampling*. In stratified sampling, the population (*i.e.* the entire set of correspondences used in the evaluation) is first divided into subpopulations, called *strata*. These strata are selected in such a way that they represent parts of the population with a common property. Useful distinctions to make when stratifying a set of correspondences are: different alignment relations (*e.g.* equivalence, subsumption), correspondences in different domains (*e.g* cats, automobiles), different expected performance of the matching system (*e.g.* hard and easy parts of the alignment), or different levels of importance to the use case (*e.g* mission critical versus nice-to-have). The strata form a partition of the entire population, so that every correspondence has a non-zero probability to end up in a sample. Then a sample is drawn from each stratum by *simple random sampling*. These samples are assessed and used to score each stratum, treating the stratum as if it were an entire population. The approximated proportion and margin of error can be calculated with *simple random sampling*.

Stratified random sampling for the evaluation of alignments has two major advantages over simple random sampling. (i) The separate evaluation of subpopulations makes it easier to investigate the conditions for the behavior of matching techniques. If the strata are chosen in such a way that they distinguish between different usages of the correspondences, we can draw conclusions about the behavior of the correspondences in a use case. For example, if a certain matching technique works very well on chemical concepts, but not on anatomical concepts, then this will only come up if this division is made through stratification. (ii) Evaluation results for the entire population acquired by combining the results from stratified random sampling are more precise than those of

simple random sampling. With simple random sampling there is always a chance that the sample is coincidentally biased against an important property. While every property that is distinguished in the stratification process will be represented in the sample.

The results of all the strata can be combined to one result for the entire population by weighing the results by the relative sizes of the strata. Let N be the size of the entire population and $N_1, \ldots, N_L$ the sizes of strata 1 to L, so that $N_1 + \cdots + N_L = N$. Then the weight of stratum h is N_h/N. Let n_h be the size of the *simple random sample* in stratum h and $\hat{P}_h$ be the approximation of proportion p in stratum h by the sample of size n_h. We do not require the sample sizes $n_1, \ldots, n_L$ to be equal, or proportional to the size of the stratum. The approximated proportion in the entire population, $\hat{P}$, can be calculated from the approximated proportions of the strata, $\hat{P}_h$, as follows:

$$\hat{P} = \frac{1}{N} \sum_{h=1}^{L} N_h \hat{P}_h$$

Due to the fact that the variance of the binomial distribution is greatest at $p = 0.5$, we know that the greatest margin-of-error occurs when $\hat{P} = 0.5$. That means that with a confidence of 95% the approximation of $\hat{P}$ lies in the interval:

$$\hat{P} \in [p - \delta, p + \delta] \quad \text{where} \quad \delta = \frac{1}{\sqrt{N}} \sqrt{\sum_{h=1}^{L} (\frac{N_h}{n_h} - 1)} \tag{2}$$

Comparative Alignment Sample Evaluation

p_A	true proportion of the correspondences produced by system A that is correct (unknown)
$\hat{P}_A$	sample approximation of p_A
$\hat{P}_{A,h}$	$\hat{P}_A$ in stratum h

To compare the performance of two systems, A and B, using sample evaluation, we calculate their respective $\hat{P}_A$ and $\hat{P}_B$ and check if their margins of error overlap. If this is not the case, we can assume with a certain confidence that p_A and p_B are different, and hence that one system is significantly better than the other. For *simple random sampling* this can be calculated as follows:

$$|\hat{P}_A - \hat{P}_B| > 2\sqrt{\frac{\hat{P}_A(1-\hat{P}_A)}{n} + \frac{\hat{P}_B(1-\hat{P}_B)}{n}} \tag{3}$$

For *stratified random sampling* this can be calculated as follows:

$$|\hat{P}_A - \hat{P}_B| > 2\sqrt{\sum_{h=1}^{L} \frac{\hat{P}_{A,h}(1-\hat{P}_{A,h})}{N} \left(\frac{N_h}{n_h} - 1\right) + \sum_{h=1}^{L} \frac{\hat{P}_{B,h}(1-\hat{P}_{B,h})}{N} \left(\frac{N_h}{n_h} - 1\right)} \tag{4}$$

For both methods the maximum difference needed to distinguish P_A from P_B with a confidence of 95% is $2/\sqrt{2n}$. So if, depending on the type of sampling performed, equation (3) or (4) holds, there is a significant difference between the performance of system A and B.

4 Alignment Sample Evaluation in Practice

In this section we will demonstrate the effects of *alignment sample evaluation* in practice by applying *stratified random sampling* on the results of the OAEI 2006 food track[10] for the estimation of Precision and we will calculate the margin of error caused by the sampling process.

The OAEI 2006 food track is a thesaurus matching task between the Food and Agriculture Organisation of the United Nations (FAO) AGROVOC thesaurus and the thesaurus of the United States Department of Agriculture (USDA) National Agricultural Library (NAL). Both thesauri are supplied to participants in SKOS and OWL Lite[11]. The alignment had to be formulated in SKOS Mapping Vocabulary[12] and submitted in the common format for alignments[13]. A detailed description of the OAEI 2006 food track can be found in [12, 13].

Five teams submitted an alignment: Falcon-AO, COMA++, HMatch, PRIOR, and RiMOM. Each alignment consisted only of one-to-one semantic equivalence correspondences. The size of the five alignments is shown below.

system	RiMOM	Falcon-AO	Prior	COMA++	HMatch	all systems
# *Found*	13,975	13,009	11,511	15,496	20,001	31,112

The number of unique *Found* correspondences was 31,112. The number of *Correct* correspondences can be estimated in the same order of magnitude. In our experience, voluntary judges can only reliably assess a few hundred correspondences per day. That means this means assessing all the *Found* correspondences in the alignments would already take many judges a few weeks of full-time work. This is only feasible with significant funding. Thus, we performed a sample evaluation.

During a preliminary analysis of the results we noticed that the performance of the different systems was quite consistent for most topics, except correspondences between taxonomical concepts (*i.e.* names of living organisms such as "Bos Taurus") with latin names where some systems performed noticeably worse than others. This was very surprising given that there was a straightforward rule to decide the validity of a taxonomical correspondence, due to similar editorial guidelines for taxonomical concepts in the two thesauri. Two concepts with the same preferred label and some ancestors with the same preferred label are equivalent. Also, when the preferred label of one concept is literally the same as the alternative label of the other and some of their ancestors have the same preferred label they are equivalent. For example, the African elephant in AGROVOC has a preferred label "African elephant" and an alternative label "Loxodonta africana". In NALT it is the other way around.

These rules allowed us to semi-automatically assess the taxonomical correspondences. This was not possible for the other correspondences. So we decided to separately evaluate correspondences from and to taxonomical concepts. We also noticed

[10] `http://www.few.vu.nl/~wrvhage/oaei2006`

[11] The conversion from SKOS to OWL Lite was provided by Wei Hu.

[12] `http://www.w3.org/2004/02/skos/mapping/spec`

[13] `http://oaei.ontologymatching.org/2006/align.html`

that most other correspondences were very easy to judge, except correspondences between biochemical concepts (*e.g.* "protein kinases") and substance names (*e.g.* "tryptophan 2,3-dioxygenase"). These required more than a layman's knowledge of biology or chemistry. So we decided to also evaluate biological and chemical concepts separately, with different judges. This led to three strata: taxonomical correspondences, biological and chemical correspondences, and the remaining correspondences. The sizes of the strata, along with the size of the evaluated part of the stratum and the corresponding stratum weights are shown below.

stratum topic	stratum size (N_h)	sample size (n_h)	stratum weight (N_h/N)
taxonomical	18,399	18,399	0.59
biological and chemical	2,403	250	0.08
miscellaneous	10,310	650	0.33
all strata	31,112	21,452	

Precision estimates using these strata have a maximum margin of error of:

$$2 \cdot \sqrt{\frac{0.5 \cdot (1-0.5)}{31112} \cdot \left(\left(\frac{18399}{18399} - 1\right) + \left(\frac{2403}{250} - 1\right) + \left(\frac{10310}{650} - 1\right) \right) \cdot 2} \approx 3.8\%$$

at a confidence level of 95%. That means that, under the assumption that there are no further biases in the experiment, a system with 82% Precision outperforms a system with 78% Precision with more than 95% confidence.

If, for example, we are interested in the performance of a system for the alignment of biological and chemical concepts and use the sample of 250 correspondences to derive the performance on the entire set of 2,403 correspondences our margin of error would be $1/\sqrt{250} \approx 6.3\%$. Comparison of two systems based on only these 250 sample biological and chemical correspondences gives results with a margin of error of $2/\sqrt{2 \cdot 250} \approx 8.9\%$. That means with a confidence level of 95% we can distinguish a system with 50% Precision from a system with 59% Precision, but not from a system with 55% Precision.

5 Conclusion

We presented two alternative techniques for the evaluation of ontology-matching systems and showed the margin of error that comes with these techniques. We also showed how they can be applied and what the statistical results mean in practice in the context of the OAEI 2006. Both techniques allow a more application-centered evaluation approach than current practice.

Apart from sampling errors we investigated in this paper, there are many other possible types of errors that can occur in an evaluation setting. (Some of which are discussed in [14].) Other sources of errors remain a subject for future work. Also, this paper leaves open the question of which technique to choose for a certain evaluation effort. For example, when you want to apply evaluation to find the best ontology matching system for a certain application. The right choice depends on which technique is more cost effective. In practice, there is a trade-off between cheap and reliable evaluation: With limited resources there is no such thing as absolute reliability. Yet, all the questions we

have about the behavior of matching systems will have to be answered with the available evaluation results. The nature of the use case for which the evaluation is performed determines which of the two approaches is more cost effective. Depending on the nature of the final application, evaluation of end-to-end performance will sometimes turn out to be more cost effective than investigating the alignment, and sometimes the latter option will be a better choice. We will apply the techniques presented in this paper to the food, environment, and library tasks of the forthcoming OAEI 2007.[14] This should give us the opportunity to further study this subject.

Acknowledgments

We would like to thank Frank van Harmelen, Guus Schreiber, Lourens van der Meij, Stefan Slobach (VU), Hap Kolb, Erik Schoen, Jan Telman, and Giljam Derksen (TNO), Margherita Sini (FAO), Lori Finch (NAL), Part of this work has been funded by NWO, the Netherlands Organisation for Scientific Research, in the context of the STITCH project and the Vitrual Laboratories for e-Science (VL-e) project.[15]

References

1. Euzenat, J., Shvaiko, P.: Ontology matching. Springer-Verlag, Heidelberg (DE) (2007)
2. Šváb, O., Svátek, V., Stuckenschmidt, H.: A study in empirical and casuistic analysis of ontology mapping results. In: Proc. of the European Semantic Web Conf. (ESWC). (2007)
3. Euzenat, J.: Semantic precision and recall for ontology alignment evaluation. In: Proc. of IJCAI 2007. (2007) 348–353
4. Ehrig, M., Euzenat, J.: Relaxed precision and recall for ontology matching. In: Proc. of K-CAP 2005 workshop on integrating ontologies. (2005) 25–32
5. Voorhees, E., Tice, D.: Building a question answering test collection. In: Proc. of SIGIR. (2000)
6. Voorhees, E.: Variations in relevance judgments and the measurement of retrieval effectiveness. In: Research and Development in Information Retrieval. (1998) 315–323
7. Meier, P.: Variance of a weighted mean. Biometrics **9**(1) (1953) 59–73
8. Hull, D.: Evaluating evaluation measure stability. In: Proc. of SIGIR 2000. (2000)
9. van Rijsbergen, C.J.: Information Retrieval. Butterworths (1979)
10. Cochran, W.G.: Sampling Techniques. John Wiley & Sons, Inc. (1977)
11. Wang, S., Isaac, A., van der Meij, L., Schlobach, S.: Multi-concept alignment and evaluation. In: Proc. of the Int. Workshop on Ontology Matching. (2007)
12. Euzenat, J., Mochol, M., Shvaiko, P., Stuckenschmidt, H., Šváb, O., Svátek, V., van Hage, W.R., Yatskevich, M.: Results of the ontology alignment evaluation initiative (2006)
13. Shvaiko, P., Euzenat, J., Stuckenschmidt, H., Mochol, M., Giunchiglia, F., Yatskevich, M., Avesani, P., van Hage, W.R., Šváb, O., Svátek, V.: Description of alignment evaluation and benchmarking results. KnowledgeWeb Project deliverable D2.2.9 (2007)
14. Avesani, P., Giunchiglia, F., Yatskevich, M.: A large scale taxonomy mapping evaluation. In: Proc. of the Int. Semantic Web Conf. (ISWC). (2005)

[14] http://oaei.ontologymatching.org/2007/

[15] http://www.vl-e.nl

Detecting Quality Problems in Semantic Metadata without the Presence of a Gold Standard

Yuangui Lei, Andriy Nikolov, Victoria Uren, and Enrico Motta

Knowledge Media Institute (KMi), The Open University, Milton Keynes,
{y.lei, a. nikolov, v.s.uren, e.motta}@open.ac.uk

Abstract. Detecting quality problems in semantic metadata is crucial for ensuring a high quality semantic web. Current approaches are primarily focused on the algorithms used in semantic metadata generation rather than on the data themselves. They typically require the presence of a gold standard and are not suitable for assessing the quality of semantic metadata. This paper proposes a novel approach, which exploits a range of knowledge sources including both domain and background knowledge to support semantic metadata evaluation without the need of a gold standard. We have conducted a set of preliminary experiments, which show promising results.

1 Introduction

Because poor quality data can destroy the effectiveness of semantic web technology by hampering applications from producing accurate results, detecting quality problems in semantic metadata is crucial for ensuring a high quality semantic web. State-of-art approaches are primarily focused on the assessment of algorithms used in data generation rather than on the data themselves. Examples include the GATE evaluation model [3], the learning accuracy (LA) metric model [2], and the balanced distance metric (BDM) model [11].

As pointed out by [5], semantic metadata evaluation differs significantly from metadata generation algorithms. In particular, the gold standard based approaches that are often used in algorithm evaluation are not suitable for two main reasons. First, it is simply not feasible to obtain gold standards from all the data sources involved, especially, when the semantic metadata are large scale. Second, the gold standard based approaches are not applicable to *dynamic* evaluation, where the process needs to take place on the fly without prior knowledge about data sources.

The approach proposed in this work addresses this issue by exploiting a range of available knowledge sources. In particular, two types of knowledge source are used. One is the knowledge sources that are available in the problem domain, including ontologies. The other type is background knowledge, which includes knowledge sources that are available globally for all applications, e.g., knowledge

sources published on the (Semantic) Web. A set of preliminary experiments have been conducted, which indicate promising results.

The rest of the paper is organized as follows. We begin in Section 2 by describing the motivation of this work in the context of a use scenario. We then present an overview of the approach in Section 3. Next in Section 4 and Section 5, we describe how to exploit each type of knowledge to support the evaluation task. We then describe the settings and the results of the experiments we carried out in this work in Section 6. Finally, we conclude with the key contributions and future work in Section 7.

2 Motivating Scenario: Ensuring High Quality for Semantic Metadata Acquisition

This work was motivated by our work on building a Semantic Web (SW) portal for KMi that would provide an integrated access to resources about various aspects of the academic life of our lab[1]. The relevant data is spread over several different data sources such as departmental databases, knowledge bases and HTML pages. In particular, KMi has an electronic newsletter[2], which describes events of significance to the KMi members. New entries are kept being added to the archive.

There are two essential activities involved in the portal, including i) extracting named entities (e.g., people, organizations, projects, etc.) from news stories in an automatic manner and ii) verifying the derived data to ensure that only data at high quality proceeds to the semantic metadata repository. Both activities take place *dynamically* on a *continuous* basis whenever new information becomes available. In particular, the involved data source is unknown to the portal prior to the metadata acquisition process. Hence, traditional gold standard based evaluation approaches are not applicable, as pre-constructing gold standards is simply not possible.

Please note that although it is drawn from the context of semantic metadata acquisition, the scenario also applies to generic semantic web applications, where evaluation often needs to be performed in an automated manner in order to filter out poor quality data dynamically whenever intermediate results are produced.

3 An Overview of the Proposed Approach

The goal of the proposed approach is to automatically detect data deficiencies in semantic metadata without having to construct gold standard data sets. It was inspired by our previous work ASDI [9], which employs different types of knowledge sources to verify semantic metadata. We extend this method towards a more powerful mechanism to support the checking of data quality by exploiting more types of knowledge sources and by addressing more types of data deficiencies.

[1] http://semanticweb.kmi.open.ac.uk
[2] http://kmi.open.ac.uk/news

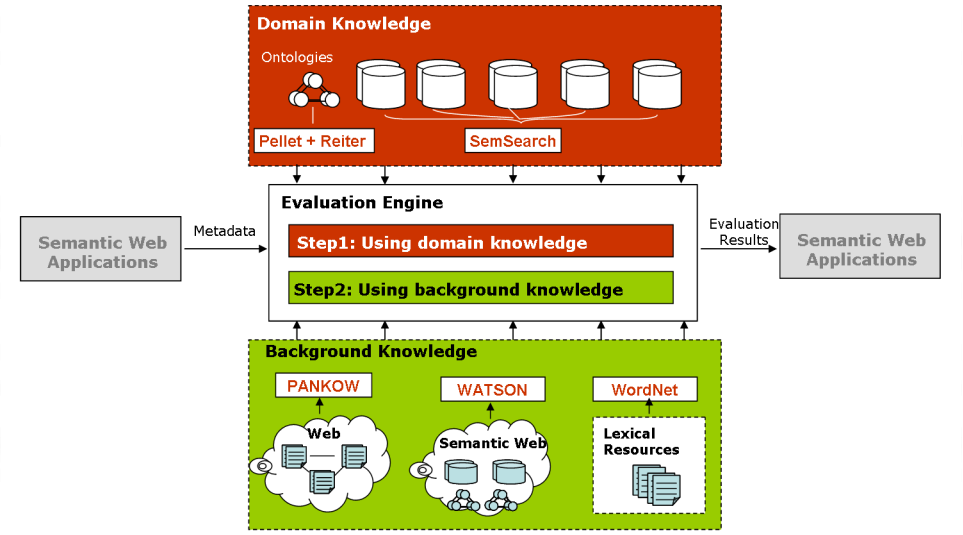


Fig. 1. An Overview of the Proposed Evaluation Approach

Figure 1 shows an overview of the proposed approach. In the following sub sections, we first describe the deficiencies addressed by the proposed approach. We then clarify the knowledge sources used in detecting quality problems.

3.1 Data Deficiencies Addressed

To clarify, we define semantic metadata as RDF triples that describe the meaning of data sources (i.e. semantic annotations) or denote real world objects (e.g., projects and publications) using the specified ontologies. In our previous work, we have developed a quality framework, called SemEval [13], which has identified a set of important data deficiencies that occur in semantic metadata, including:

- **Incomplete annotation**, which defines the situation where the mapping from the objects described in the data source to the instances contained in the semantic metadata set is not exhaustive.
- **Inconsistent annotation**, which denotes the situation where entities are inconsistent with the underlying ontologies. For example, an organization ontology may define that there should be only one director for an organization. The inconsistency problem occurs when there are two directors in the semantic metadata set.
- **Duplicate annotation**, which describes the deficiency in which there is more than one instance referring to the same object. An example situation is that the person *Clara Mancini* is annotated as two different instances, for example *Clara Mancini* and *Clara*.

– **Ambiguous annotation**, which expresses the situation where an instance of the semantic metadata set can be mapped back to more than one real world object. One example would be the instance *John* (of the class *Person*) in the context where there are several people described in the same document who have the name.
– **Inaccurate annotation**, which defines the situation where the object described by the source has been correctly picked up but not accurately described. An extreme scenario in this category is *mis-classification*, where the data object has been successfully picked up and been associated with a wrong class. An example would be the *Organization* instance *Sun Microsystem* marked as a person.
– **Spurious annotation**, which defines the deficiency where there is no object to be mapped back to for an instance. For example, the string *"Today"* annotated as a person.

The proposed approach is designed to address all these data deficiencies except the first one. This is because the approach concentrates especially on the quality status of the semantic annotations that are *already* contained in the given semantic metadata set.

3.2 Knowledge Sources Exploited

As shown in Figure 1, two types of knowledge sources are exploited to support the evaluation task, namely *domain knowledge* and *background knowledge*.

Domain knowledge. Three types of knowledge sources are often available in the problem domain: i) domain ontologies, which model the problem domain and offer rules and constraints for detecting conflicts and inconsistencies contained in the evaluated data set; ii) semantic data repositories, which contain facts of the problem domain that can be looked upon to examine problems like inaccuracy, ambiguity, and inconsistency; and iii) lexical resources, which contain domain specific lexicons that can be used to link the evaluated semantic metadata with specific domain entities. As will be detailed in Section 4, domain knowledge is employed to detect inconsistency, duplicate, ambiguous and inaccurate annotation problems.

Background knowledge. The knowledge sources that fall into this category include: i) online ontologies and data repositories, ii) online textual resources, and iii) general lexical resources. The first two types of knowledge sources are exploited to detect possible deficiencies that might be associated with those entities that are not included in the problem domain (i.e., those entities that do not have matches). General lexical resources, on the other hand, are employed to expand queries when finding matches of the evaluated entity.

Compared to domain knowledge, one characteristic of background knowledge is that it is generic and is available to all applications. Another important feature of the knowledge, especially the first two types of background knowledge, is that they are less trustworthy than domain specific knowledge as the (semantic) web is an open environment where anyone can contribute. Corresponding to the two

types of knowledge sources exploited, the deficiency detection process comprises two major steps, which are described in the following sections.

4 Detecting Data Deficiencies Using Domain Knowledge

The tasks involved in this step are centered around the detection of four types of quality problems that are common to semantic metadata, namely inconsistent, duplicate, ambiguous, and inaccurate problems. The process starts with the detection of inconsistencies that may exist between the evaluated semantic metadata entity with the data contained in the specified semantic data repositories. It then investigates the duplicate problem using the same annotation context. The third task involved is detecting ambiguous and inaccurate problems by querying the available semantic data repositories.

Detecting inconsistencies. Please note that we are only interested in data inconsistencies at the ABox level. Such inconsistencies may be caused by disjointness axioms or the violation of property restrictions. First, disjointness leads to inconsistency when the same individual belongs to two disjoint classes at the same time. For example, the annotation "Ms Windows is a Person" is inconsistent with the statement that defines it as a technology, as the two classes are disjoint with each other. Second, violation of property restrictions (e.g., domain/range restriction, cardinality restriction) also causes inconsistencies. For example, if the ontology defines that there should be only one director in an organization, there is an inconsistency if two people are classified as director.

To achieve the task of inconsistency detection, we employ ontology diagnosis techniques. Each inconsistency is represented by a so-called minimal inconsistent subontology (MISO) [7], which includes all statements and axioms that contribute to the conflict. An OWL-reasoner with explanation capability is able to return a MISO for the first inconsistency found in the data set. The process starts with locating a single inconsistency using the Pellet OWL reasoner [8]. It then discovers all the inconsistencies by using Reiter's hitting set tree algorithm [12], which builds a complete consistent tree by removing each ABox axiom from the MISO one by one. Please see [12] for the detail.

Detecting duplicate problems. This task is achieved by seeking matches of the evaluated entity within the same annotation context, i.e., within the values of the same property of the same instance that contains the evaluated entity. For example, when evaluating the annotation *(story x, mentions-person, enrico)*, the proposed approach examines other person entities mentioned in the same story for detecting the duplicate problem. Domain specific lexicons are used in the process (e.g., the string "OU" stands for "Open University") to address domain specific abbreviations and terms.

Detecting ambiguous and inaccurate problems. This task is fulfilled by querying the available data repositories. When there is more than one match found, the evaluated entity is considered to be ambiguous, as its meaning (i.e., the mapping to real world data objects) is not clear. For example, in the case of evaluating the person entity "John", there is more than one match found in the KMi domain

repository. The meaning of the instance needs to be disambiguated. In the situation where there is an inexact match, the entity is computed as inaccurate. As to the third possibility where there is no match found, the proposed approach turns to background knowledge to carry out further investigation. We used SemSearch [10], a semantic search engine, to query the available data repositories, and a suite of string matching mechanisms to refine the matching result.

5 Checking Entities Using Background Knowledge

There are three possibilities when matches could not be found for the evaluated entity in the problem domain. One is that the entity is correct but not included in the problem domain (e.g., IBM, BBC, and W3C with respect to the KMi domain). The second possibility is *mis-classification*, where the entity is wrongly classified, e.g., "Sun Microsystems" as a "person". The third one is *spurious annotation*, in which the entity is erroneous, e.g., "today" as a "person". Hence, this step focuses on detecting two types of quality problems: mis-classification) and spurious annotation.

The task is achieved by computing possible classifications using knowledge sources published on the (semantic) web. The process begins by querying the semantic web. If satisfactory evidence cannot be derived, the approach then turns to textual resources available on the web (i.e., the general web) for further investigation. If both attempts fail, the system considers the evaluated entity *spurious*.

We used i) *WATSON* [4], a semantic search tool developed in our lab, to seek classifications of the evaluated term from the semantic web; and ii) *PANKOW* [1], a pattern-based term classification tool, to derive possible classifications from the general web. *Detecting mis-classification problems* is achieved by comparing the derived classifications (e.g., *company* and *organization* in the case of evaluating the annotation "Sun Microsystems as person") to the type of the evaluated entity (which is the class *person* in the example) by exploring domain ontologies and general lexicon resources like WordNet [6]. In particular, the disjointness of classes are used to support the detection of the problem. General lexicon resources are also exploited to compute the semantic similarities of the classifications.

6 Experiments

In this work we have carried out three preliminary experiments, which investigate the performance of the proposed approach in the KMi domain. In the following subsections, we first describe the settings and the methods of the experiments. We then discuss the results of the experiments.

6.1 Setup

The experimental data were collected from the previous experiment carried out in ASDI [9], in which we randomly chose 36 news stories from the KMi news

archive[3] and constructed a gold standard annotation collection by asking several KMi researchers to manually mark them up in terms of *person, organization* and *projects*. We used the semantic metadata set that was automatically gathered from the chosen news stories by the named entity recognition tool ESpotter [14] as the data set that needs evaluation. We then experimented with this semantic metadata set using a gold standard based approach and the proposed approach.

In order to get a better idea of the performance of the proposed approach on employing different types of knowledge sources, we conducted three experiments: the first experiment used the constructed gold standard annotation collection; the second one used domain knowledge sources; and the third experiment used both domain knowledge and background knowledge. In particular, for the purpose of minimizing the influences that may be caused by other factors such has human intervention, we developed automatic evaluation mechanisms for both the gold standard based approach and the proposed approach, which use the same matching mechanism. Table 1 shows the results, with each cell presenting the total number of the correspondent data deficiencies (i.e., row) found in the data set with respect to the extracted entity type (or the the sum of all types).

Table 1. The Data Deficiency Detection Results of the Experiments

Deficiency	**People**	**Organizations**	**Projects**	**Total**
Experiment 1: Using the gold standard annotations				
Incomplete annotation	17	16	9	42
Inconsistent	**n/a**(not applicable)			
Duplicate	3	10	0	13
Ambiguous	0	1	0	1
Inaccurate	0	1	0	1
Spurious	8	17	0	25
*Experiment 2: Using **domain knowledge only***				
Incomplete annotation	**n/a**			
Inconsistent	1	0	0	1
Duplicate	3	10	0	13
Ambiguous	0	1	0	1
Inaccurate	1	3	0	4
Spurious	33	45	2	80
*Experiment 3: Using **both domain knowledge and background knowledge***				
Incomplete annotation	**n/a**			
Inconsistent	**5**	**8**	**0**	**13**
Duplicate	3	10	0	13
Ambiguous	0	1	0	1
Inaccurate	1	3	0	4
Spurious	**5**	**8**	**0**	**13**

[3] http://kmi.open.ac.uk/news

6.2 Discussion

Assessing the performance of the proposed approach is difficult, as it largely depends on three factors, including i) whether it is possible to get hold of good data repositories that cover most facts of the problem domain, ii) whether the relevant topics have gained good publicity on the (semantic) web, and iii) whether the background knowledge itself is of good quality and trustworthy. Here we compare the results of the different experiments in the hope of finding some clues of the performance.

Comparing the proposed approach with the gold standard based approach. As shown in the table, the performances on detecting duplicate, ambiguous and inaccurate problems are quite close. This is because that, like gold standard annotations, the KMi domain knowledge repositories cover all the facts (including people, projects, organizations) that are contained in the domain. On the other hand, there are two major differences between the gold standard based approach and the proposed approach.

One major difference is that, in contrast with the gold standard based approach, the proposed approach is able to detect inconsistent annotations but with no support for the incomplete annotation problem. This is because the proposed approach deliberately includes domain ontologies as a type of knowledge sources and does not have the knowledge of full set of annotations of the data source.

Another major difference lies in the detection of the spurious annotation problem. More specifically, there is a big difference between the first experiment and the second one. This is mainly caused by the fact that many entities extracted from the news stories are not included in the domain knowledge (e.g., "IBM", "BBC", "W3C"), and thus are not being to be covered by the second experiment. But they are contained in the manually constructed gold standard.

There is also a significant difference between the first experiment and the third one with respect to the detection of spurious annotations. Further investigation reveals two problems. One is that the gold standard data set is not perfect. Some entities are not included but correctly picked up by the extraction tool. "EU Commissioner Reding as a person" is such an example. The other problem is that background knowledge can sometimes lead to false conclusions. On the one hand, some spurious annotations are computed as correct, due to the difficulties in distinguishing different senses of the same word in different contexts. For example, "international workshop" as an instance of the class *Organization* is computed as correct, whereas the meaning of the word *organization* when associating with the term is quite different from the meaning of the class in the KMi domain ontology. On the other hand, false alarms are sometimes produced due to the lack of publicity of the evaluated entity in the background knowledge. For example, in the KMi SW portal, the person instance *Marco Ramoni* is computed as spurious, as not enough evidence could be gathered to draw a positive conclusion.

Comparing the performance of the approach between using and without using background knowledge. With 12 inconsistencies discovered and 58 spurious prob-

lems cleared among the 80 spurious problems detected in the second experiment, the use of background knowledge has proven to be effective in problem detection in the KMi domain. This is mainly because the relevant entities that are contained in the chosen news stories collection have gained fairly good publicity. "Sun Microsystem", "BBC" and "W3C" are such examples. As such, classifications can be easily drawn from the (Semantic) web to support the deficiency detection task. However, as described above, we have also observed that several false results have been produced by the proposed approach.

In summary, the results of the experiments indicate that the proposed approach works reasonably well for the KMi domain when considering zero human effort is required. In particular, domain knowledge is proven to be useful in detecting those problems that are highly relevant to the problem domain, such as ambiguous and inaccurate annotation problems. The background knowledge, on the other hand, is quite useful for investigating those entities that are outside.

7 Conclusions and Future Work

The key contribution of this paper is the proposed approach, which, in contrast with existing approaches that typically focus on the evaluation of semantic metadata generation algorithms, pays special attention to the quality evaluation of semantic metadata themselves. It addresses the major drawback of current approaches suffered when applying to data evaluation, which is the need for gold standards, by exploiting a range of knowledge sources.

In particular, two types of knowledge source are used. One is the knowledge sources that are available in the problem domain, including domain ontologies, domain specific data repositories and domain lexical resources. They are used to detect quality problems of those semantic metadata that are contained in the problem domain, including data inconsistencies, duplicate, ambiguous and inaccurate problems. The other type is background knowledge, which includes ontologies and data repositories published on the semantic web, online textual resources, and general lexical resources. It is mainly used to detect quality problems that are associated with those data that are not contained in the problem domain, including mis-classification and spurious annotations.

We have conducted three preliminary experiments examining the performance of the proposed approach, with each focusing on the use of different types of knowledge sources. The study shows encouraging results. We are, however, aware of *a number of issues* associated with the proposed approach. For example, real time response is crucial for dynamic evaluation, which takes places at run time. How to speed up the evaluation process is an issue that needs to be investigated in the future. Another important issue is the impact of the trustworthiness of different types of knowledge on the evaluation.

Acknowledgements

This work was funded by the X-Media project (www.x-media-project.org) sponsored by the European Commission as part of the Information Society Technologies (IST) programme under EC grant number IST-FP6-026978.

References

1. P. Cimiano, S. Handschuh, and S. Staab. Towards the Self-Annotating Web. In *Proceedings of the 13th International World Wide Web Conference*, pages 462 – 471, 2004.
2. P. Cimiano, S. Staab, and J. Tane. Acquisition of Taxonomies from Text: FCA meets NLP. In *Proceedings of the ECML/PKDD Workshop on Adaptive Text Extraction and Mining*, pages 10 – 17, 2003.
3. H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL02)*, 2002.
4. M. d'Aquin, M. Sabou, M. Dzbor, C. Baldassarre, L. Gridinoc, S. Angeletou, and E. Motta. WATSON: A Gateway for the Semantic Web. In *4th European Semantic Web Conference (ESWC'07)*, 2007.
5. K. Dellschaft and S. Staab. Strategies for the Evaluation of Ontology Learning. In P. Buitelaar and P. Cimiano, editors, *Bridging the Gap between Text and Knowledge - Selected Contributions to Ontology Learning and Population from Text*. IOS Press, December 2007.
6. C. Fellbaum. *WORDNET: An Electronic Lexical Database*. MIT Press, 1998.
7. P. Haase, F. Harmelen, Z. Huang, H. Stuckenschmidt, and Y. Sure. A framework for handling inconsistency in changing ontologies. In *Proceedings of the International Semantic Web Conference (ISWC2005)*, 2005.
8. A. Kalyanpur, B. Parsia, E. Sirin, and B. Grau. Repairing Unsatisfiable Concepts in OWL Ontologies. In *3rd Eupopean Semantic Web Conference*, pages 170–184, 2006.
9. Y. Lei, M. Sabou, V. Lopez, J. Zhu, V. S. Uren, and E. Motta. An Infrastructure for Acquiring High Quality Semantic Metadata. In *Proceedings of the 3rd European Semantic Web Conference*, 2006.
10. Y. Lei, V. Uren, and E. Motta. SemSearch: A Search Engine for the Semantic Web. In *Proceedings of the 15th International Conference on Knowledge Engineering and Knowledge Management(EKAW2006)*, 2006.
11. D. Maynard, W. Peters, and Y. Li. Metrics for Evaluation of Ontology-based Information Extraction. In *Proceedings of the 4th International Workshop on Evaluation of Ontologies on the Web*, Edinburgh, UK, May 2006.
12. R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95, 1987.
13. Enrico Motta Yuangui Lei, Victoria Uren. SemEval: A Framework for Evaluating Semantic Metadata. In *The Fifth International Conference on Knowledge Capture (KCAP 2007)*, 2007.
14. J. Zhu, V. Uren, and E. Motta. ESpotter: Adaptive Named Entity Recognition for Web Browsing. In *Proceedings of the Professional Knowledge Management Conference*, 2004.

Tracking Name Patterns in OWL Ontologies

Vojtěch Svátek, Ondřej Šváb

University of Economics, Prague, Dep. Information and Knowledge Engineering,
Winston Churchill Sq. 4, 130 67 Praha 3, Prague, Czech Republic
svatek@vse.cz, svabo@vse.cz

Abstract. Analysis of concept naming in OWL ontologies with set-theoretic semantics could serve as partial means for understanding their conceptual structure, detecting modelling errors and assessing their quality. We carried out experiments on three existing ontologies from public repositories, concerning the consistency of very simple name patterns—subclass name being a certain kind of parent class name extension, while considering thesaurus relationships. Several probable taxonomic errors were identified in this way.

1 Introduction

Concept names in semantic web (OWL) ontologies with set-theoretic semantics are sometimes viewed as secondary information. Indeed, for logic-based reasoners, which are assumed to be the main customers exploiting these ontologies, anyhow cryptic URLs can serve well. Experience however shows that even in ontologies primarily intended for machine consumption, the naming policy is almost never completely arbitrary. It is important for ontology developers (and maintainers, adoptors etc.) to be able to see the semantic structure of a large part of the ontology at once, and ontology editors normally use base concept names (local URLs) and not additional linguistic labels within their taxonomy view. At the same time, while inspecting possibly complex OWL axioms, self-explaining concept names (even independent of their context in the taxonomy) are extremely helpful.

This leads us to the hypothesis that concept naming in OWL ontologies can (at least in some cases) be a useful means for analysing their conceptual structure, detecting modelling errors and assessing their quality. Obviously, a 'true' evaluation of concept naming in specialised domain ontologies requires deep knowledge of the domain. We however assume that even in specialised ontologies, the 'seed' terms often belong to generic vocabulary and the domain specialisation is rather achieved via adding syntactic attributes (such as adjectives or nouns in apposition), leading to multi-word terms. The class-subclass pairs would then often be characterised by the presence of a common token (or sequence of tokens) on some particular position; we can see this as a simple (atomic) *name pattern*. Although the proportion of instances of such a pattern only represent a fraction of all subclass relationships[1], in large- and medium-sized ontologies this may suffice for partial assessment of the consistency of naming, as part of ontology quality evaluation.

[1] Based on our preliminary analysis, we estimate this fraction to float around 50%, depending on domain specificity and other factors.

Atomic name patterns can then be weaved into more complex pattern structures with their own semantics. The deeper understanding of the structure of an ontology thus acquired can help in e.g. mapping it properly to other ontologies.

The paper is structured as follows. Section 2 sets up the token-level background for our name patterns. Section 3 explaines the name patterns themselves. Section 4 describes the initial experiments on three ontologies and attempts to interpret their results. Section 5 surveys some related work. Finally, section 6 summarises the paper and outlines some future work.

2 Matching Tokens in OWL Concept Names

All name patterns we consider in this work are built upon the *sub-string relationships* between pairs of concept names, at token level. The token-level relationship can in general have the nature of prefix, postfix or infix, possibly adjusted with some connective. For example, the name 'WrittenDocument' can be extended via prefix to 'HandWrittenDocument' or via infix to 'WrittenLegalDocument'. A postfix extension could be e.g. 'WrittenDocumentTemplate', which, however, unlike the previous ones, would not be adequate for a subclass of 'WrittenDocument', as the main term (distinguished by its placement as end token) has changed. An adequate postfix extension for a subclass would in turn be e.g.'WrittenDocumentWithComments'; here however the postfix has the form of prepositional construction appended to the main term (thus preserved).

Tokenisation is, for 'technical' items such as OWL concept names, usually assumed to rely on the presence of one of a few delimiters, in particular: underscore (`Concept_name`), hyphen (`Concept-name`) and change of lowercase letter to uppercase (`ConceptName`), which is most parsimonious and therefore most frequent. Although the semantics of these delimiters could in principle differ (especially the hyphen is likely to be used for more specific purposes than the remaining two, on some occasions), we will treat them as equivalent for the sake of simplicity. We will also ignore sub-string relationship without explicit token boundary (i.e. between two single-word expressions), assuming that they often deviate from proper subclass relationship (as in 'fly' vs. 'butterfly', or even worse e.g 'stake' vs. 'mistake').

The mentioned token-level structures then have to be tracked over the *ontology structure* (for simplicity let us only consider taxonomic paths). This could lead to an inventory of *naming patterns*, some of which we considered in our start-up analysis presented below. The most obvious naming pattern is of course the one already mentioned: a subclass name being token-level extension of its parent class. Such patterns can already be assigned some status wrt. ontology content evaluation and possible refactoring. Although the 'token analysis' approach used is admittedly quite naive from the NLP point of view, we believe that, due to the restricted nature of concept names in ontologies, we would not need much more for covering the majority of multi-word names in real-world ontologies.

3 Some Ammunition for Pattern-Based Evaluation

Let us now outline a few, still rather vague, initial hypotheses concerning the interpretation of name patterns.

The first one, concerning subclassing, is central in our initial investigation:

Hypothesis 1 *If the main term in the name of a class and the main term in the* multi-word *name of its immediate subclass do not* correspond[2] *then it is likely that there is a conceptual incoherence.*

The hypothesis anticipates that ontology designers should not often, while subclassing, substantially change the meaning of the main term in the name, as the main term is likely to denote the conceptual type of the underlying real-world entity, and they are obliged to keep the set-theoretic consistency (all instances of the subclass also have to be instances of the parent class). They may however subclass a multi-word name with a rather specific single-word name.

The second hypothesis is closely related:

Hypothesis 2 *If the two main terms from Hypothesis 1 only correspond via some long-range terminological link then it is likely that there is a shift to a more specific domain with its own terminology.*

This hypothesis might help suggesting points for breaking large monolithic ontologies into more and less specific parts.

We also formulated two hypotheses that involve more extensive graph structures of the taxonomy.

Hypothesis 3 *Concept with the same main term in their names should not occur in separate taxonomy paths.*

In other words, if there are several partial taxonomies with the same main term, they are candidates for merger.

Hypothesis 4 *If two taxonomy paths exist such that one contains a class X and its subclass Y, and the other contains a class Z and its subclass W, such that the name of X is token-level extension of the name of Z,* with different main term*, and the name of Y is token-level extension of the name of W,* with different main term*, then both paths should be linked with some property and the name pattern should probably apply for the descendants of Y and W as well.*

This amounts to identification of 'parallel' taxonomies of related (but conceptually different) entities, which may also be quite important e.g. in ontology refactoring as well as mapping.

In the experiments below we only systematically compare Hypothesis 1 to our findings. We however occasionally mention the other three hypotheses where relevant.

[2] The specification of 'correspondance' is discussed in section 4.1.

4 Experiments

In the initial, manual[3], phase of our experiments, we restricted the analysis to 3 small- to medium-sized ontologies we picked from public repositories. Their choice was more-or-less 'random', we however avoided ontologies that appear as mere (converted) ad hoc taxonomies without the assumption of set-theoretic semantics, as well as 'toy' models designed for demonstrating DL reasoning (such as 'pizzas' or 'mad cows'), which are actually quite common in such repositories, cf. [9].

4.1 Settings

In designing the experiments, there were numerous choices, especially concerning:

1. What *patterns* to follow
2. Whether to only consider the own structure of the ontology or also that of *imported* ontologies such as upper-level ones (namely, SUMO, in two out of the three cases)
3. Whether to require for fulfilling the patterns that the main term should be identical in the parent class and subclass, or also allow *hyponymy/synonymy*.

For the first issue we eventually decided to only consider two concrete patterns. The one is the presence of a common *end token*; note that this covers all cases of prefix and infix extension. The second (which proved much more rare) is the postfix extension starting with the 'of' preposition.

For the second issue we decided to restrict the analysis to the *current ontology* only (i.e. both members of the evaluated concept pairs had to be from the current ontology), but including concepts from imported ontologies that belong to the same domain (or mean only very slight domain generalisation). The rationale is that we did not intend to evaluate the way the concepts from the current ontology are grafted on the upper-level ontology, but only the design of the current ontology proper.

For the third issue, we decided to use *WordNet*[4], with the assumption that a general thesaurus is likely to contain the main terms of multi-word domain terms. However, we separately counted and listed the cases where the pattern compliance was established via WordNet only. We did not use WordNet for *single-token* subclass terms[5]; we rather excluded them from the analysis.

The results of the analysis amount to the simple statistics of:

1. Class-subclass pairs where (one of the two considered) name patterns hold directly.
2. Class-subclass pairs where a name pattern holds via WordNet only.
3. Class-subclass pairs where name patterns don't hold even via WordNet, but we eventhough assessed the subclass relationship as correct.
4. Class-subclass pairs where name patterns don't hold even via WordNet, and we assessed the subclass relationship as incorrect (at least at the level of class names).

[3] For examining the ontologies, we simply unfolded their taxonomies in Protégé.

[4] `http://wordnet.princeton.edu/`

[5] Our main focus are specialised domain ontologies, whose single-token terms are likely to either miss in standard lexical databases or exhibit a meaning shift there.

In the tables below, the cases 2, 3 and 4 are explicitly listed and commented. Three symbolic labels were added for better overview. ○ means: correct relationship, contradicts our Hypothesis 1. ● means: incorrect, conforms to our Hypothesis 1. Finally, ⊗ means: main terms correspond via thesaurus, i.e. Hypothesis 1 does not apply[6].

The number of cases 3 ('false positives') and 4 ('true positives') can be viewed as evaluation measures for our envisaged method of conceptual error detection. There could potentially be 'false positives' even among the cases 2 (and theoretically even among the cases 1) due to homonymy of terms; we however did not clearly identify any such case. The *accuracy* of our approach can thus be simply established as the ratio of the number of cases 4 vs. the number of cases 3+4.

4.2 ATO Mission Models Ontology

This, US-based military (ATO probably stands for 'Air Tasking Order') ontology, which we picked from the DAML repository[7], is an ideal example of highly specific ontology rich in multi-token names; there are very few single-token ones, and none of these is involved as subclass in one of the subclass relationships. The ontology contains 86 classes (aside classes inherited from imported ontologies), and there are 116 immediate subclass relationships[8] (including some multiple inheritance). Of them, 95 comply with the name patterns, and 21 don't. Table 1 lists and comments the subclass relationships that break the name pattern. We assume (see the table) that the majority of non-compliance cases (11, i.e. 52%) are modelling errors[9]; some others (5, i.e. 24%) are not strict non-compliance as relationship between the names could be determined using WordNet, and only a few (5, i.e. 24%) seem to be 'false alarms'. In addition, the ontology contains some portions relevant to Hypotheses 3 (e.g. some 'missions' placed beyond the main 'mission' taxonomy and under some other concepts) and 4 (e.g. parallel taxonomies for 'missions' and 'mission plans').

4.3 Government Ontology

This ontology (also from the DAML repository), is relatively smaller and less domain-specific; it contains 53 classes (aside classes inherited from imported ontologies), and there are 27 immediate subclass relationship (including some multiple inheritance). Of the subclass relatioships, 11 comply with the name patterns and 13 don't; finally, 3 involve a single-token subclass, thus being irrelevant for our method. Table 2 lists and comments the subclass relationships that break the name patterns.

4.4 EuroCitizen Ontology

This ontology, picked from the *OntoSelect*[10] repository, contains 71 classes. It has no explicit imports, but largely borrows from SUMO at higher levels of the taxonomy. It

[6] But Hypothesis 2 might do if the correspondence is 'long range' only.

[7] `http://www.daml.org/ontologies/`

[8] Here we also considered relationships such that the superclass belonged to the imported but tightly thematically linked ATO ontology.

[9] Or, possibly, artifacts of the DAML→OWL conversion.

[10] `http://olp.dfki.de/ontoselect/`

Superclass	Subclass/es	Comment
AirspaceControlMeasure	AirCorridor TimingReferencePoint DropZone CompositeAirOperationsRoute	● Subclassing indeed looks misleading. A 'measure' can be *setting up* e.g. a corridor, but not the corridor *itself*.
AirStation	AirTankerCellAirspace	● Rather evokes *part-of* relationship but hard to judge w/o domain expertise.
ATOMission	AircraftRepositioning	○ By the available comment, means AircraftRepositioningMission. However, 'repositioning' looks like acceptable term, though not hyponym of 'mission' in WordNet.
ATOMission	CompositeAirOperations	⊗ 'Mission' is direct hyponym of 'operation' in WordNet. Note however the misuse of plural form.
ATOMissionPlan	IndividualLocationReconnaissanceRequestMission MissileWeaponAttackMission	● The 'Plan' token erroneously missing. The remaining 19 sibling subclasses do have it.
CommandAndControlProcess	AirborneElementsTheaterAirControlSystemMission	● Subclass clearly misplaced, 'mission' concept non contiguous.
CommandAndControlProcess	ForwardAirControl	● Probably means ForwardAirControlProcess.
CommandAndControlProcess	FlightFollowing	⊗ 'Following' could be seen as process (it is hyponym of 'processing' in WordNet). Hypothesis 2 might apply.
ConstraintChecking	RouteValidation	○ Specialisation to subdomain; 'validation' should be closely related to 'checking' but surprisingly is not in WordNet.
ControlAgency	ForwardAirControllerAirborne	○ A tricky case: the end token in subclass is actually an attribute of the true entity ('controller'). Furthermore, although the relationship between 'agency' and 'controller' is not intuitive, it might be OK in the domain context.
ForwardAirControl	AirborneBattleDirection	⊗ 'Direction' is direct subclass of 'control' in WordNet.
GroundTheaterAirControlSystem	ControlAndReportingCenter ControlAndReportingElement	○ Though the relationship between the end tokens is not intuitive, it looks OK in the domain context.
IntelligenceAcquisition	AirborneEarlyWarning	● Rather looks like two subsequent processes: warning is *preceded* by intelligence acquisition. However the end token 'acquisition' bears little meaning by itself.
ModernMilitaryMissile	ArmyTacticalMissileSystem	● A system (i.e. group) of missiles, possibly including a launcher, is probably not a subclass of 'missile'.
PrepositionedMaterielTask	GroundStationTankerMission	⊗ 'Mission' is close hyponym of 'task' in WordNet.
SupportingTask	GroundStationTankerMission	⊗ As above.

Table 1. Name pattern breaks in the ATO Mission Models ontology

Superclass	**Subclass/es**	**Comment**
AreaOfConcern	TransnationalIssue	○ Pattern 2 applies. Interestingly, here the 'semantic' term is rather that *after* 'of': 'issue' is close hyponym of 'concern' in WordNet. Note that Hypothesis 3 would incorrectly suggest to integrate this concept into the taxonomy of geographic areas.
DiplomaticOrganization	ConsulateGeneral	○ A tricky case: the subclass name is a noun phrase obeying French rather than English syntax rules.
GovernmentOrganization	GovernmentCabinet	⊗ 'Cabinet' is hyponym of 'organisation' in WordNet. Hypothesis 2 might apply.
JudicialOrganization	AppealsCourt (+ 3 other court types)	⊗ 'Court' is hyponym of 'organisation' in WordNet. Hypothesis 2 might apply.
LegislativeOrganization	LegislativeChamber	○ Correct. None of the senses of 'chamber' is closely related to 'organisation' in WordNet
OverseasArea	BritishCrownColony UnincorporatedUnitedStatesTerritory	⊗ Both 'colony' and 'territory' are close hyponyms of 'area' in WordNet.
PoliticalParty	PoliticalCoalition	● Political coalitions often have similar rights as parties but they are not conceptually identical. 'Coalition' is also *not* hyponym nor synonym of 'party' in WordNet.
SuffrageLaw	RestrictedSuffrage	● The (restricted) suffrage by itself is obviously different from the law that imposes it.
SuffrageLaw	VoterAgeRequirement	⊗ With some reservation, voter age requirements can probably be viewed as 'suffrage laws'. This case however reveals the pitfalls of using WordNet, as 'requirement' is indeed *hyponym* of 'law' there. Hypothesis 2 might apply.

Table 2. Name pattern breaks in the Government ontology

Superclass	Subclass/es	Comment
Blood	BloodGroup	● Incorrect. In the veins there are not amounts of a bloodgroup but amounts of blood *having* some group.
CombatSport	MartialArt	○ Correct. Somewhat marginal usage of 'art'.
ContentBearingObject	NaturalLanguage	⊗ 'Object' is again an extremely versatile concept; but 'natural language' is its long-range hyponym in WordNet. Hypothesis 2 might apply.
HumanAttribute	ReligiousBelief	○ Correct. The problem is due to the notion of 'attribute' being extremely versatile.
HumanBloodGroup	RhesusBloodGroupSystem	● Incorrect. The Rhesus system is an individual rather than class; it *defines* blood groups rather than having them as instances.
LandArea	StateOrProvince	⊗ 'Province' is direct hyponym of 'area' in WordNet. However, the term should not be treated as multi-word proper; it is a logical disjunction.
Region	GeographicArea	⊗ 'Area' is direct hyponym of 'region' in WordNet.
TeamSport	IceHockey	⊗ 'Hockey' is hyponym of 'sport' in WordNet. Hypothesis 2 might apply.
WaterSport	InTheWater OnTheWater	● Shortcut that makes the names too context-dependent.

Table 3. Name pattern breaks in the EuroCitizen ontology

is rather heterogeneous (with respect to its relatively tiny size), but contains clusters of related concepts, where name patterns can be identified. The overall quality of the ontology does not seem to be very high, as it contains many clear modelling errors, such as apparent instances formalised as classes. The outcomes of analysis are in Table 3.

4.5 Summary

Table 4 shows the overall figures. The results are obviously most promising for the ATO Mission Models ontology, which is most domain-specific of the three. In general, the proportion of multi-word names seems to decrease with the growing generality of the ontology (EuroCitizen being the most general of the three). The *accuracy* of 'inconsistency alarms', if they were properly implemented, could be acceptable for human inspection and evaluation of the ontology. However, perhaps with the exception of ATO Mission Models, the *coverage* of our simple approach is still too small to guarantee substantial 'cleaning' of taxonomic errors.

	ATO Missions	Government	EuroCitizen
Subclass relationships	116	27	62
with multi-token subclass	116	24	40
Pattern-compliant (identical)	95	11	30
Pattern-compliant (WordNet)	5	8	4
Pattern-non-compliant, incorrect ('true alarm')	11	2	4
Pattern-non-compliant, correct ('false alarm')	5	3	2
Pattern proportion (w/o use of WordNet)	82%	41%	48%
Accuracy of 'alarm'	69%	40%	67%

Table 4. Summary of results

5 Related Work

Our research is to some degree similar to projects aiming at converting shallow models such as thesauri or directory headings to more structured and conceptually clean ontologies [2–6]. The main difference lays in our assumption that the ontologies in question are already intended to bear set-theoretical semantics, and that the 'inconsistencies' in naming patterns are due to either sloppy naming (possibly just reflecting shortcut terminology used by domain practitioners) or more serious modelling errors, rather than being an inherent feature of (shallow) models.

On the other hand, the research in 'true' OWL ontology evaluation and refactoring has typically been focused on their logical aspects [1, 10]. Our research is, in a way, parallel to theirs. We aim at similar long-term goals, such as detecting potential modelling inconsistencies or making implicit structures explicit. We however focus on a different aspect of ontologies: the naming policy. Due to the subtler nature of consistency or implicit structures in these realms (usually requiring some degree of acquaintance with the domain), the conclusions of name pattern analysis have probably to be more cautious than those resulting from logic-based analysis.

6 Conclusions and Future Work

We presented a simple method of tracking name patterns (based on token-level extensions) over OWL ontology taxonomies, which could help detect some errors with respect to their set-theoretic interpretation. Initial experiments on three ontologies from public repositories indicated that the method has some potential, although the performance will probably largely vary from one ontology to another, especially with respect to their domain specificity.

There are various directions in which our current work ought to be extended. First of all, the so far manual process of pattern (non-compliance) detection used in the very first experiments should be replaced by an *automatic* one. We also plan to *reuse* experience from popular NLP-oriented methods of ontology 'reconstruction' from shallow models, such as those described in [3] or [5]. Consequently, we should, analogously to those approaches, adopt at least a simple *formal model*. Furthermore, concept names used as identifiers are obviously not the only lexical items available in ontologies. future

(especially, more automated) analysis should pay similar attention to additional, potentially even multi-lingual *lexical labels* (based on `rdf:label`) and *comments*, which may help reveal if the identifier name is just a shortcut of the 'real' underlying concept name. In addition to class names, *property* naming (in connection with their domain and range) should also be followed, e.g. as drafted in [7]. In long term, we perceive as important to combine the analysis of naming patterns with the analysis of *logical patterns*, in the sense of 'guessing' the modeller's original intention that got distorted due to the representational limitations of OWL. Our closely related interest is also the use of discovered patterns for mapping *between* ontologies. We already started to test the behaviour of some well-known (string-based and graph-based) ontology mapping methods with respect to naming patterns present in ontologies, using synthetic ontology-like models [8]. In the future, the analysis of (naming and other) patterns would be used as pre-processing step to mapping.

The research was partially supported by the IGA VSE grants no.12/06 "Integration of approaches to ontological engineering: design patterns, mapping and mining", no.20/07 "Combination and comparison of ontology mapping methods and systems", and by the Knowledge Web Network of Excellence (IST FP6-507482).

References

1. Baumeister J., Seipel D.: Smelly Owls – Design Anomalies in Ontologies. In: Proc. FLAIRS 2005, 215–220.
2. Giunchiglia F., Marchese M., Zaihrayeu I.: Encoding Classifications into Lightweight Ontologies. In: Proc. ESWC 2006.
3. Hepp M., de Bruijn J.: GenTax: A Generic Methodology for Deriving OWL and RDF-S Ontologies from Hierarchical Classifications, Thesauri, and Inconsistent Taxonomies. In: Proc. ESWC 2007.
4. Kavalec M., Svátek V.: Information Extraction and Ontology Learning Guided by Web Directory. In: ECAI Workshop on NLP and ML for ontology engineering. Lyon 2002.
5. Magnini B., Serafini L., Speranza M.: Making Explicit the Hidden Semantics of Hierarchical Classifications. In: Proc. AI*IA 2003.
6. Serafini L., Zanobini S., Sceffer S., Bouquet P.: Matching Hierarchical Classifications with Attributes. In: Proc. ESWC 2006.
7. Svátek, V.: Design Patterns for Semantic Web Ontologies: Motivation and Discussion. In: 7^{th} Conf. on Business Information Systems (BIS-04), Poznan, April 2004.
8. Šváb O., Svátek V.: In Vitro Study of Mapping Method Interactions in a Name Pattern Landscape. Accepted to the Ontology Matching (OM-07) workshop at ISWC 2007, Busan, Korea.
9. Tempich C., Volz R.: Towards a benchmark for Semantic Web reasoners - an analysis of the DAML ontology library. In: EON Workshop at ISWC 2003.
10. Vrandecic D., Sure Y.: How to Design Better Ontology Metrics. In: Proc. ESWC 2007.

The 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference

November 11~15 2007
BEXCO, Busan KOREA